1999-05

An analysis procedure for advanced propulsor design

Renick, Dirk Hampton.

Monterey California. Naval Postgraduate School

An Analysis Procedure for Advanced Propulsor Design

by

Dirk Hampton Renick

Submitted to the Departments of Ocean Engineering and Mechanical Engineering
in partial fulfillment of the requirements for the degrees of

Naval Engineers Degree

and

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1999

An Analysis Procedure for Advanced Propulsor Design

by
Dirk Hampton Renick

Submitted to the Departments of Ocean Engineering and Mechanical Engineering
on May 18, 1999, in partial fulfillment of the
requirements for the degrees of
Naval Engineers Degree
and
Master of Science in Mechanical Engineering

Abstract

A propeller which operates in the shear flow found near the aft end of marine vehicles experiences an intimate coupling between the propeller's induced velocity field and the rotational inflow. The presence of the propeller's induced velocity field causes the inflow to accelerate, which redistributes the vorticity present in the inflow. This redistribution causes a change in the nominal propeller inflow. Because the propeller now experiences a different inflow, the propeller induced velocity field is altered. Thus, there is an intimate coupling between the vorticity present in the fluid inflow and the propeller-generated induced velocity field.

Lifting surface propeller blade design codes are incapable of analytically representing the vortical interaction between the induced velocity field and the rotational inflow. An additional difficulty is encountered as downstream blade elements, which would be present in a multi-component propulsor, pass through the singularity wake sheets shed by upstream components. For these reasons, the propeller blade design code should be coupled with an external flow solver which is capable of transporting vorticity.

Previous researchers have coupled propeller blade design codes with Reynolds Averaged Navier Stokes (RANS) flow solvers. This powerful method made possible multi-element propulsor design in the presence of rotational inflow. Unfortunately, the use of a RANS code is costly in terms of time and computational resources.

This thesis focuses on coupling a propeller blade design code with an axisymmetric, multi-element through-flow code, developed by Drela. The throughflow code uses an integral boundary layer method to solve for the boundary layer flow, and a streamline curvature formulation to solve for the inviscid, outer flow. The main advantage of the present method over previous methods is an order of magnitude reduction in computation time. Validation cases were performed to validate the various components of the coupling procedure, as well as the coupling methodology as a whole. A design case is presented which shows the use of this methodology in design.

Thesis Supervisor: Justin E. Kerwin
Title: Professor of Naval Architecture

Acknowledgments

This thesis and the underlying research would not have been possible without the unwavering, solid support of my wife Naoko, who worried with me every step of the way, offered encouragement when needed, and lay my head gently to rest every night for these three years. My parents gave me the tools and the background to succeed. My friends always showed that there was a way other than my own. My mentors showed me the broader paths. My past teachers taught me to think.

Professor Justin Kerwin, Dr. Todd Taylor, Rich Kimball and Gerard McHugh provided invaluable insight and guidance into the sometimes black art of propeller blade design and the nuances of coupling a propeller blade design code with a throughflow solver. Their insightful answers to my steady barrage of questions led to the successful completion of this thesis.

Contents

List of Figures

List of Tables

Symbology

A	stream tube cross sectional area
A_1, A_2	area vector normal to streamtube volume along streamtube
B	number of blades
B^-, B^+	area vector normal to streatube volume normal to streamtube
H	Boundary Layer shape parameter
M	mach number
n	Surface normal vector
n	propeller rotation rate (rps)
P	Pressure
Q	mass flow rate
r	local radius from centerline
r_c	meridional radius of curvature of streamtube
R_T	Total resistance
$R\epsilon_\theta$	Boundary Layer momentum thickness-based Reynold's number
s	unit vector in direction of streamtube
T	Barehull resistance
T_θ	circumferential blade thickness
u_e	boundary layer edge velocity
$\vec{V}_{eff}$	Effective Inflow Velocity
$\vec{V}_{induced}$	Induced Inflow Velocity
$\vec{V}_{nominal}$	Nominal Inflow Velocity
$\vec{V}_{total}$	Total Inflow Velocity
$\vec{V}_r$	Radial component of meridional velocity
$\vec{V}_x$	Axial component of meridional velocity
V_θ	Meridional tangential velocity
$\vec{V}_\theta^*$	Circumferential mean induced tangential velocity
v_m	meridional velocity magnitude
v_θ	tangential (swirl) velocity
W_T	Taylor wake fraction
W_N	Nominal volumetric wake fraction
Z	number of propeller blades
δ	boundary layer thickness
δ^*	boundary layer displacement thickness
ΔS	added entropy
ΔW	work addition
γ_B	bound vorticity
Γ	blade circulation
Ω	blade rotation rate
ρ	local cell density
Π	streamtube normal pressure
θ	boundary layer momentum thickness
ν	kinematic viscosity
∇	Gradient Operator

δ	Boundary Layer thickness
δ^*	Boundary Layer displacement thickness
ρ	density
κ	Radius of curvature of a streamtube
$\vec{\Gamma}$	Circulation
Γ_B	Bound circulation
θ	Boundary Layer momentum thickness
ω	Rotation rate
$\vec{\omega}$	vorticity
$\vec{\omega}_\theta$	Tangential vorticity

Chapter 1

This Thesis in the Propeller Design Process

A propeller which operates in the shear flow found near the aft end of marine vehicles experiences an intimate coupling between the propeller's induced velocity field and the rotational inflow. The presence of the propeller's induced velocity field causes the inflow to accelerate, which redistributes the vorticity present in the inflow [14]. This redistribution causes a change in the total inflow velocity to the propeller. Because the propeller now experiences a different inflow, the propeller induced velocity field is altered. Thus, there is an intimate coupling between the vorticity present in the fluid inflow and the propeller-generated induced velocity field. To design more efficient, and more complex, propulsor geometries, it is necessary to accurately account for this physical interaction phenomena.

A potential, vortex lattice lifting surface propeller blade design code such as Kerwin's Propeller Blade Design (PBD) code is incapable of analytically representing the vortical interaction between the induced velocity field and the rotational inflow. An additional difficulty is encountered as downstream blade elements, which would be present in a multi-component propulsor, pass through the singularity wake sheets shed by upstream components. For these reasons, the propeller blade design code should be coupled with a throughflow fluid solver which is capable of modelling vorticity transport.

This thesis focuses on coupling a propeller blade design code developed at MIT by Kerwin [5] with an axisymmetric, multi-element throughflow code, developed at MIT by Drela [3]. Drela's throughflow solver uses an integral boundary layer method to solve for the boundary layer flow, and a streamline curvature formulation to solve for the inviscid, outer flow. The main advantage of the streamline curvature throughflow code is an order of magnitude reduction in computation time compared with the RANS coupling methodology [14].

1.1 Fluid Velocity Terminology

Before proceeding further, it is necassary to first clear up what the propeller designer means when talking about fluid velocities. The nominal inflow is the fluid velocity in the region of the propeller with the propeller not operating. The total inflow is the fluid velocity with the propeller operating. The total inflow velocity is composed of an induced velocity, which is induced by the presence of the propeller blade and trailing wake singularity distributions, and an effective inflow.

$$\vec{V}_{total} = \vec{V}_{induced} + \vec{V}_{effective} \tag{1.1}$$

Figure 1-1 graphically shows the relative magnitudes and directions of the total, induced, and effective inflow velocities for a typical propeller.

While the nominal inflow is the inflow field when there is no propeller present, it is not equal to the effective inflow. This is due to the aforementioned interaction effects between the propeller

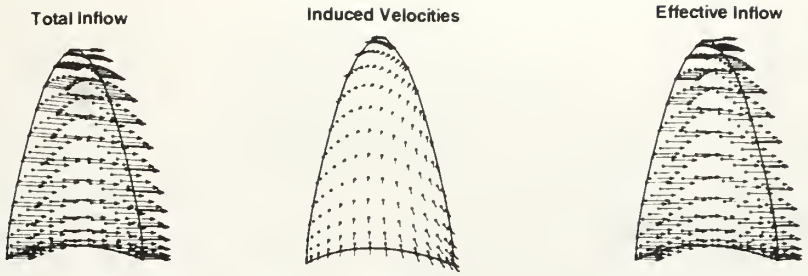


Figure 1-1: The total, induced, and effective inflow velocity fields generated by Propeller 4119, J=0.833, in inviscid flow.

vortex lattice and the vorticity in the inflow. This can be seen from the definition of the vorticity vector.

$$\vec{\omega} = \vec{\nabla} \times \vec{V} \tag{1.2}$$

A non-radially-uniform meridional inflow suggests the presence of a tangential component of vorticity.

$$\vec{\omega}_\theta = \frac{\partial \vec{V}_r}{\partial x} - \frac{\partial \vec{V}_x}{\partial r} \tag{1.3}$$

The tangential vorticity can be thought of as a ring which circumscribes the propeller shaft [18]. The propeller induced velocity causes a contraction of the tangential vorticity ring. As this ring contracts, Kelvin's thereom requires that the strength of the vorticity remain constant. Therefore, from equation (1.3), it is evident there is a change in $\frac{\partial V_x}{\partial r}$ to counteract the change in $\frac{\partial V_r}{\partial x}$. If there were no tangential vorticity present in the inflow, though, the nominal and effective inflows are exactly equal. This fact is used later to validate the coupling methodology in this thesis.

1.2 Effective Inflow

All propeller design and analysis takes place in the effective inflow. Recall that the effective inflow is the total inflow with the effects of the propeller (*i.e.*, the propeller induced velocity component) removed. The effective inflow is a non-physical flow field. It can not be measured in a water tunnel or behind a ship in operation. And, yet, determining the effective velocity field is critical to the success of the propeller design process.

Historically, to determine the effective inflow velocity, a model of the ship is constructed, and velocity profile surveys are taken in the propeller disk plane of the unappended model. This is the nominal inflow. The effective inflow velocity is taken as a fraction of the nominal inflow.

$$\vec{V}_{eff} = \left(\frac{1 - W_T}{1 - W_N} \right) \vec{V}_{nom} \tag{1.4}$$

As shown above, though, this linear scaling across all radii means that in the hub region, where the boundary layer dominates the inviscid flow, the velocities are too high, and the velocities over the rest of the blade span are too low. In fact, this method produces the correct velocity at only one radius along the propeller span!

A more modern method is to computationally compute the flow in the region of the propeller by solving the equations of fluid motion, accounting for the presence of the propeller. Thus, the exact total inflow velocity is known at every radii across the propeller span. Once the induced velocity is subtracted from the total inflow velocity, the effective inflow is known.

Besides the effective inflow velocity, the propulsor designer also requires an accurate knowledge of the thrust power which must be generated by the propulsor at the design speed.

K_T, K_Q, η
Propeller Diameter
Spanwise Circulation, $\Gamma(r)$
Propeller RPM
Cavitation Index σ

Table 1.1: Outputs from a lifting line analysis.

1.3 The Coupled Hull Flow Resistance Problem

To accurately design a propeller requires an accurate knowledge of the ship's resistance. The increased fluid velocity along the stern of the vessel due to the presence of the operating propeller causes an increase in drag since there is more incomplete pressure recovery. The vortical interaction also wreaks havoc upon the traditional tow tank resistance calculation, since, for vessels with high aft prismatic coefficients, the presence of the propeller can inhibit boundary layer separation.

Historically, the values for the resistance of the ship with and without a propeller operating are related through the thrust deduction coefficient.

$$R_T = (1 - t)\, T \tag{1.5}$$

Modern computation techniques which model the interaction of the propeller and the fluid flow around the hull solve both the effective inflow and thrust deduction problems by nearly exactly solving for the fluid flow over the hull with the propeller operating. In this method, the entire submarine and duct, if present, is modelled in the computer. The shear flow is exactly computed along the body, and the correct propeller interaction effects are also modelled.

Once the effective velocity field is known, and the ship, or submarine, resistance is known, we can get on with the task of designing a propeller.

1.4 Modern Design Techniques

The starting point for the propeller design process is the effective inflow. The designer then uses a lifting line theory to optimize the gross propeller characteristics. In lifting line theory the blades are represented as two dimensional "sticks" of bound vorticity [13]. Refinements upon the basic lifting line model allow for hub and duct modelling, and multiple blade rows [2]. Bowles' advanced lifting line method [2] also allows for the cavitation and strength considerations in his lifting line development. Not included in this analysis are skew and rake effects. The outputs of the lifting line analysis shown in Table 1.1 give the several gross propulsor characteristics.

Once the lifting line analysis is complete, a lifting surface propeller code is used to find the two dimensional blade shapes, pitched about a raked and skewed genarator line, which produce the required circulation $\Gamma(r)$. While further discussion of lifting-surface theory is delayed until Section 1.5, there are several key output parameters from lifting surface design.

In the serial progression from lifting line to lifting surface design shown in Figure 1-2, the effective inflow is assumed constant. However, a quick check of the numerically predicted induced velocity components against the empirically assumed induced velocity components (remember, the designer had to scale a physical total inflow to get to the effective inflow by subtracting the induced velocity components) would show that there are differences. This should drive the propeller designer to rederive a new effective inflow and optimize a new propeller. Another tact is to couple the preliminary design tools with a viscous (*i.e.*, able to model the viscosity in the governing fluid equations of motion) or inviscid Euler throughflow solver which can accurately model the effects of the propeller vorticity upon the global flow solution. The requirement for preliminary design, however, is a fast, computationally "cheap" solution to allow proper exploration of the design space. The current RANS coupling is not fast in that sense. As will be shown later, an axisymmetric streamtube solver meets both requirements.

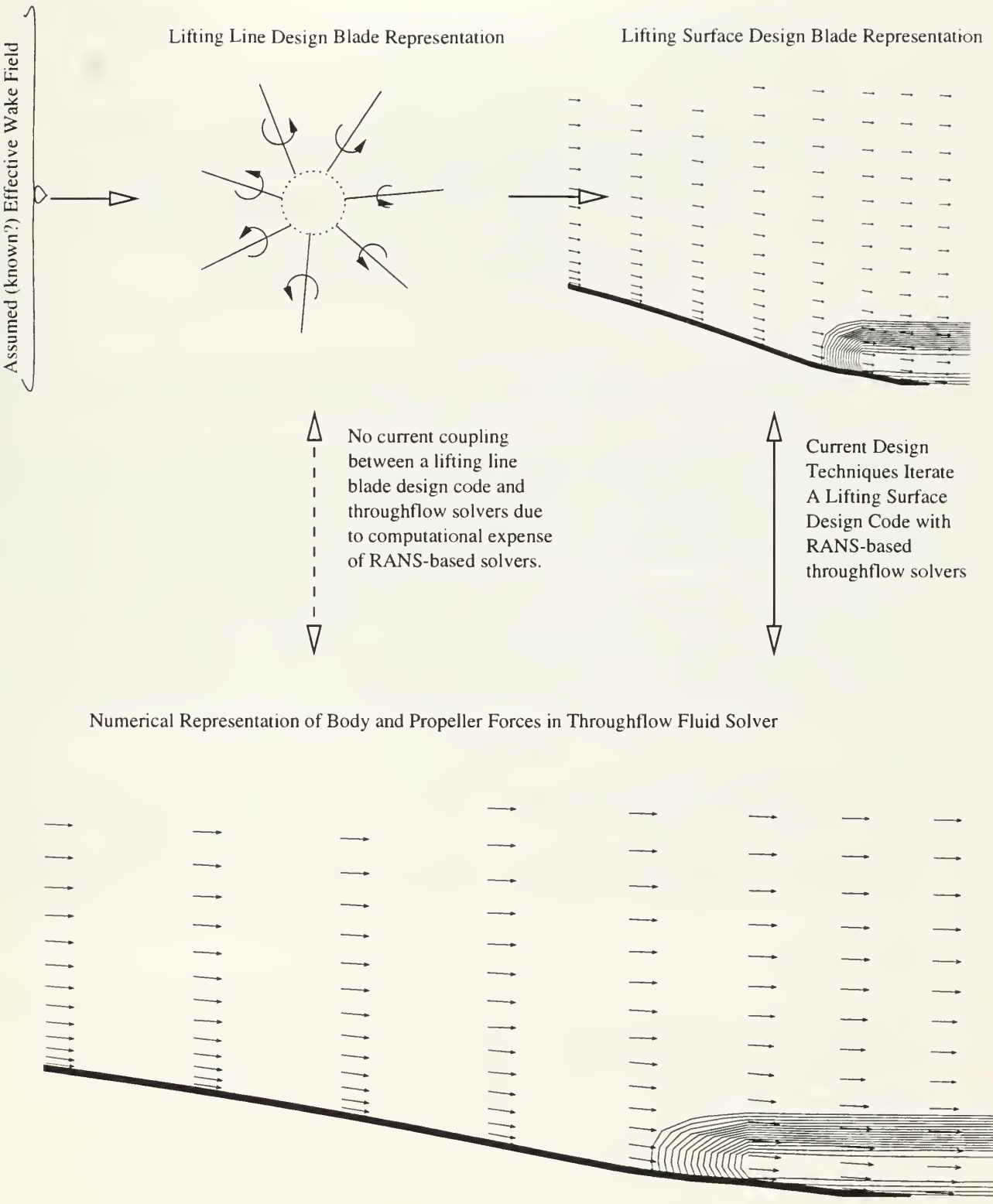


Figure 1-2: The prelimary propeller design process relies upon an assumed effective wake (*i.e.*,velocity field) as input to a lifting line analysis to optimize the propeller and lifting surface analysis to design the three dimensional blade shapes. There is a limited amount of coupling with throughflow solvers to iterate upon the effective wake.

Performance	K_T, $10K_Q$, η, to check thrust and torque.
Blade Offsets	Expressed in the traditional propeller design parameters of rake, skew, pitch, camber, and thickness distributions this specifies the geometry of the blade for construction.
Spanwise Circulation	The final circulation distribution may not match the original desired circulation distribution due to three dimensional rake and skew effects.
Local Pressure Coefficient	This gives and indication of cavitation.

Table 1.2: Outputs from a lifting surface analysis.

1.5 Lifting-Surface Propeller Blade Design and Analysis

In a traditional propeller lifting surface propeller code, a grid lattice is placed on the blade mean camber surface, the hub and duct (if present) and the trailing wake system [15]. Each lattice segment is assigned a strength of vorticity. On the solid body surfaces, such as the blade and hub, a control point is placed near the center of the grid lattice. The strength of each vortex lattice segment is assigned by satisfying the kinematic boundary condition that the flow must be tangent to the propeller, hub, and duct surfaces at every control point.

Mathematically, the propeller problem involves a simple matrix equation. By attacking the geometry of the problem, an influence matrix is formulated which gives the velocity induced by a unit strength of vorticity along every vortex lattice upon every control point – an $[INF]$luence matrix. When multiplied by the actual vortex segment strengths, the induced velocity at every control point is known.

$$[INF]\left[\vec{\Gamma}\right] = \vec{V}_{induced} \tag{1.6}$$

/noindent To model the effects of blade thickness, a souce distribution is placed coincident with the blade vortex lattice system.

Next, to satisfy the kinematic boundary condition, the physics of the problem dictate that the component of the induced velocity normal to the blade, when added to the component of the effective inflow velocity normal to the blade must be zero.

$$\left[[INF]\left[\vec{\Gamma}\right] + \vec{V_{eff}}\right] \cdot n = 0 \tag{1.7}$$

Equation (1.7) is the heart of the propeller lifting surface code.

A lifting surface propeller code can be used for either the design of a new propeller or an analysis of an exsisting propeller. In the case of a propeller design, a radial loading distribution is prescribed, and a blade shape is found which produces the desired loading by manipulating the geometry of the blades. In blade analysis, the unique strength of the vortex lattices segments is found which satisfies equation (1.7). From equation (1.7) it is seen that an incorrect effective inflow velocity will give either an incorrect blade shape, or an incorrect prediction of blade loading.

The propeller forces resulting from the vorticity and source distributions are calculated from the Kutta-Joukowski and Lagally theorems, respectively. A Lighthill leading edge suction force correction is applied to these forces, and the propeller's sectional drag is calculated either based on strip wise two dimensional empirical drag coefficients, or a stripwise 2-D integral boundary layer calculation.

The propeller blade analysis code used in this study is an extension of a previously reported lifting surface design code [12, 5]. The extensions replace the image hub and duct with vortex lattice

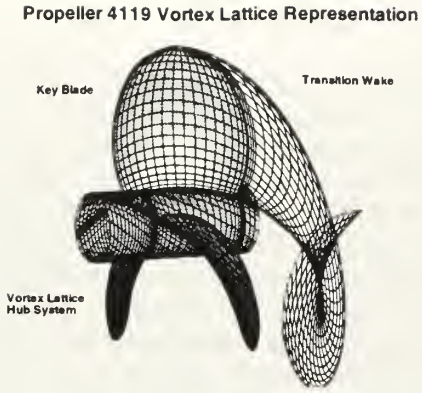


Figure 1-3: 3 bladed propeller 4119 on a straight shaft. Notice that the hub discretization extends upstream of the hub. The inner radius of the transition wake vortex lattice is the innermost hub streamline.

representations of the hub and duct [14, 17]. The computer representation of the blade and hub vortex lattice, and transition wake is shown in Figure 1.5.

The end result of the propeller blade analysis routine is the complete circumferential mean induced velocity over the blade surface. The circulation, $\vec{\Gamma}_B$,is computed from the circumferential mean tangential induced velocity by the following relationship which follows from the Euler turbine equation.

$$\vec{\Gamma}_B(r) = \frac{-2\pi r V_\theta^*}{Z} \tag{1.8}$$

Equation (1.8) is at the heart of the present coupling technique with the streamline curvature method. In the streamline curvature method, as in turbomachinery mean line design, the designer is most interested in determining the amount of turning, or swirl, being applied to the flow through the action of the blades. Therefore, in the coupling with the streamline curvature method, the swirl $r(\Delta V_\theta)$ is directly calculated by multiplying V_θ^* by the local radius r. The use of the swirl velocity to couple between the propeller code and the throughflow solver is unique to the present research. In previous coupling methodologies, the propeller is represented as distributed body forces within the RANS throughflow solver.

1.6 Coupling with a Viscous Throughflow Solver

The major shortcoming of the lifting surface formulation described above is the inability to model the vorticity interaction. As previsouly mentioned, the propeller induction velocities cause a redistribution of the vorticity present in the inflow. This interaction is critical in the design of modern, full stern submarines, where the stern prismatic coefficient is so high that the propulsor actually inhibits separation of the boundary layer.

The second area of propulsor design affected by this shortcoming is in the design of multi-element propulsors. The problem is that at points where the trailing wake vorticity from the upstream blade rows intersect the control points of the downstream blades, the mathematics of the problem produces a singular solution. Multi-element propulsor design is important in the design of modern submarine propulsors, and in the design of waterjet pumps.

And yet, the lifting surface formulation is unequalled by any other technique in its ability to rapidly and accurately predict the forces produced by a propeller. But it can not model vorticity interaction. Modern throughflow solvers, though, can accurately model vorticity interaction, but are too computationally expensive to model the whole propeller problem in three dimensions. Modelling the three dimensional propeller blade problem in a fast vortex-lattice lifting-surface code means

that a computationaly axisymmetric "cheap" viscous throughflow solver can be used to solve for the circumferential mean flow over the body.

Past researchers have coupled propeller design codes with axisymmetric RANS throughflow solvers. This method is extremely robust, but also extremely time and computationally intensive. Thus, it seems logical to couple the propeller blade design code with a streamline curvature code, because the streamline curvature code runs extremely quickly. A simple test case, such as a three bladed propeller on a straight shaft in inviscid flow, takes around 12 hours to run on a typical work station. More complex cases, such as a submarine propulsor, take on the order of days. As a comparison, it takes less than 15 minutes to run the simple propeller in inviscid flow with the present streamline curvature coupling method. Obviously long computer run times do not leave much time for the engineer to refine and optimize the design since so much time is taken up by the analysis of a single design point. The potential time reduction would seem well worth the effort of incorporating the streamline curvature method into the preliminary design as it would allow for more exploratory design efforts, rather than computational analysis.

Chapter 2

Flow Theory

2.1 Computational Solution to Fluid Flow Problems

The governing equations of motion for any fluid flow are the Navier-Stokes Equation [22], written here in compact vector form.

$$\frac{D\vec{v}}{Dt} = -\frac{1}{\rho}\nabla P + \nu\nabla^2\vec{v} + \vec{\nabla}H \tag{2.1}$$

The solution to Equation (2.1) requires discretization in time and space. To ensure accuracy and stability of the numeric solution, the discretization sizes are directly related to the smallest length and time scales present in the flow. It can be shown that due to the extremely small length and time scales involved in boundary layer flows, a complete, numeric solution to the Navier-Stokes equations would require computational resources far in excess of those available today. Therefore, various levels of approximations are introduced to find solutions to equation (2.1).

To make the Navier-Stokes equation tractable, one fundamental approximation is to ignore the viscous terms. This results in the Euler equations of motion for an inviscid fluid. Hydrodynamicists are able to simplify the problem further by considering the fluid as incompressible. For extremely simple flow cases, there exist numerous, classical theoretical solutions to the incompressible, inviscid flow equations. The value of these classical results is not to be underestimated in that they provide an irrefutable answer by which numeric solution techniques can be validated.

Solutions to real world flow problems are only possible using numerical schemes. The large success of incompressible, inviscid numeric computer codes shows that even with the inviscid and incompressible approximations, the majority of the physics of the flow is still correctly modelled. There are, however, ways to introduce the effects of the viscosity without the computational expense of solving the fully three dimensional Navier-Stokes equations.

Since viscous effects are mostly confined to a small region of the flow domain, an effective approximation is to use a "divide-and-conquer" strategy. In this technique, an inviscid Euler solver is used to solve the majority of the flow domain, and a boundary layer solver is used to solve for the boundary layer growth along a solid wall. The two codes co-exist by matching the boundary conditions (pressure and velocity) at the interface between the two flow regions. This type of coupling is a displacment body type of scheme and is the scheme employed by the numeric codes used in this thesis. The basic idea is to use a fast, inviscid solver to solve for the inviscid flow and a fast boundary layer routine to solve for the the boundary layer flow.

2.1.1 The Boundary Layer Equations

A significant approximation to the full Navier Stokes equation can be made through the application of dimensional analysis [25]. The resulting simplfied equations, usually termed the Thin Shear Layer Equations or the Boundary Layer equations are quite simplified indeed. They are written here with

the Conservation of Mass equation for completeness in the subsequent derivation of the Integral Boundary Layer Equations.

$$\text{Continuity} \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.2}$$

$$\text{Momentum} \quad \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \approx \frac{\partial u_e}{\partial t} + u_e\frac{\partial u_e}{\partial x} + \frac{1}{\rho}\frac{\partial \tau}{\partial y} \tag{2.3}$$

To formulate the so-called integral boundary layer equations, Equation (2.2) is pre-multiplied by the quantity $(u - u_e)$ and subtracted from Equation (2.3) [25]. The resulting equation is then integrated over the domain for the boundary to infinity.

$$\frac{\tau_w}{\rho} = \frac{\partial}{\partial t}\int_0^\infty (u_\epsilon - u)dy + \frac{\partial}{\partial x}\int_0^\infty u(u_e - u)dy + \frac{\partial u_e}{\partial x}\int_0^\infty (u_\epsilon - u)dy - u_e v_w \tag{2.4}$$

where

τ	shear stress on the solid boundary
u_ϵ	inviscid edge velocity
$u(y)$	local velocity
v_w	velocity through the wall, usually termed as wall transpiration

The key at this point is to recognize the definition of the displacement thickness and momentum thickness buried beneath some algebra within Equation (2.4). Recall the following equations are the definitions for the displacement and momentum thicknesses.

$$\text{Displacement Thickness} \quad \delta^* = \int_0^\infty (1 - \frac{u}{u_e})dy \tag{2.5}$$

$$\text{Momentum Thickness} \quad \theta = \int_0^\infty \frac{u}{u_e}(1 - \frac{u}{u_e})dy \tag{2.6}$$

Inserting Equations (2.5) and (2.6) into Equation (2.4) gives the common form of the integral boundary layer equations.

$$\frac{\tau}{\rho U^2} = \frac{C_f}{2} = \frac{1}{u_e^2}\frac{\partial}{\partial t}(u_e\delta^*) + \frac{d\theta}{dx} + (2\theta + \delta^*)\frac{1}{u_e}\frac{du_e}{dx} - \frac{v_w}{u_e} \tag{2.7}$$

Ignoring the wall transpiration term, and allowing only steady flows, Equation (2.7) reduces still further.

$$\frac{C_f}{2} = \frac{d\theta}{dx} + (2 + H)\frac{\theta}{u_e}\frac{du_e}{dx} \tag{2.8}$$

where $H = \frac{\delta^*}{\theta}$, momentum shape factor

From a computation standpoint, Equations (2.7) and (2.8) are superior because they are parabolic in nature and can be solved through a marching routine. That is, starting from a given upstream starting condition, the solution downstream is found by marching downstream. Equation (2.3), on the other hand, is hyperbolic and must be solved globally and, usually, iteratively.

There remains, however, the complication of a closure relation. In equation (2.8) there are more unknowns than equations. Therefore, we must use an auxiliary equation which relates one, or more, of the unknowns to the others. This auxiliary equation is termed a closure relation when applied to boundary layer flows, and is usually based upon a mix of empiricism and physics. The closure relations used within MTFLOW and the PBD-MTFLOW coupling codes are those derived by Swafford [23].

2.2 Streamline Curvature Method

The streamline curvature method is a powerful method for solving for the inviscid flowfield within a channel or annular passage [1, 22]. Because this method is inviscid, it must be coupled with a boundary layer routine to solve for the boundary layer flow. The two then must operate in tandem since the boundary layer tends to displace the inviscid streamlines off the body by the displacement thickness, which causes a redistribution of the inviscid streamlines.

The throughflow computer code developed by Drela [3] is based on the conservative formulation of the steady state Euler eqations developed by Drela [4]. This approach is an improvement upon the original Streamline Curvature Method formulation [19]. Section 2.2.1 shows the basics of the streamline curvature method, section 2.2.2 outlines Drela's reformulation of the problem in a finite volume sense, and section 2.2.3 shows the MTFLOW computer code implementation, which makes another intelligent simplification.

The streamline curvature method has been previously applied to hydrodynamic open propulsors [20].

2.2.1 The Original Streamline Curvature Method

The streamline curvature method is applicable for axisymmetric meridional flow problems. It was first derived to solve internal flow problems. In the original streamline curvature method, the steady-state conservation of momentum equation is formulated which accounts for changes in radial momentum due to meridional curvature, fluid acceleration, and tangential velocity, or swirl, effects [11].

Fluid accelerations in the radial direction are due to

1. Acceleration in the Streamtube Direction.

$$v_m(\frac{\partial v_m}{\partial s})$$

which has an outward directed radial component

$$v_m(\frac{\partial v_m}{\partial s})\sin\phi$$

2. Meridional Curvature of the Streamtube. The centripetal acceleration is

$$\frac{v_m^2}{r_c}$$

3. Tangential Velocity. A tangential velocity component (*i.e.*, swirl) implies a centripetal acceleration

$$\frac{v_\theta^2}{r}$$

The streamline curvature method is based upon the solution of the equations of motion along a streamline in steady flow. In a streamline and normal coordinate system (s , n) the equations of motion are given as the sum of the three acceleration effects cited above[1].

$$-\frac{\partial P}{\partial s} + \rho g_s = \rho V_m \frac{\partial V_m}{\partial s} - \rho\omega \frac{\partial r V_\theta}{\partial s} \tag{2.9}$$

$$-\frac{\partial P}{\partial n} + \rho g_n = \rho V^2 \kappa \tag{2.10}$$

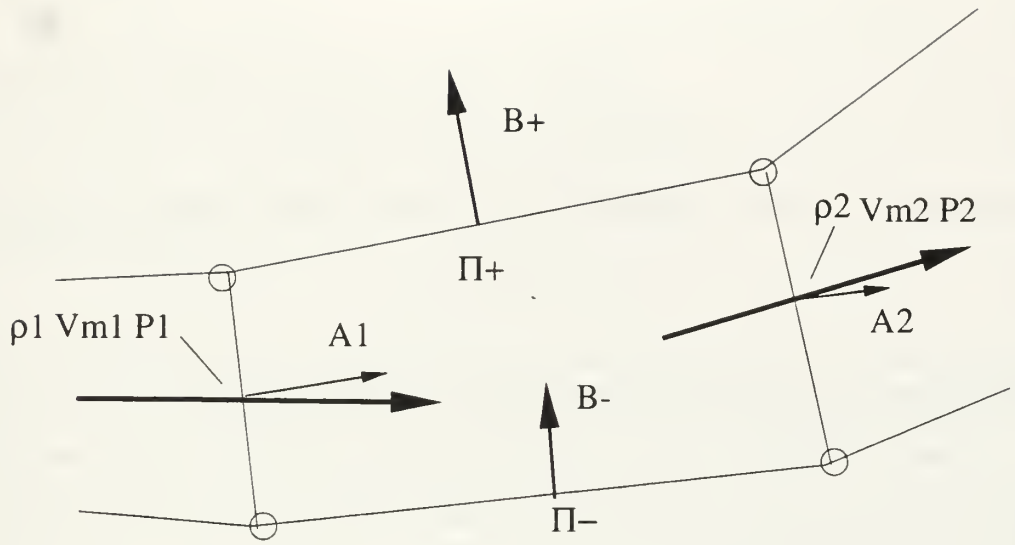


Figure 2-1: Finite Volume geometry and definitions for conservation equations.

Notice that the integral of equation (2.9) along the streamline reduces to the usual Bernoulli equation when the rate of rotation, ω, is zero.

Equation (2.10) is usually called the radial equilibrium equation. The position of the streamtube boundaries directly gives the fluid radius of curvature term κ. This simple ODE is integrated in the radial direction to derive the mean velocity within the streamtube. The resulting error in the conservation of mass equation is used to drive a relaxation procedure to alter the streamlines towards their correct positions.

The main advantage of the streamline curvature method over traditional time-marching methods is that it is orders of magnitude faster, and hence, applicable to industrial design problems [4]. The main disadvantage of the streamline curvature routine is that the iterative solvers lack numerical stability in regions of supersonic flow, and do not correctly position shocks within the flow. The method of Drela [4] was originally developed to correct this deficiency and correctly treat shocks.

2.2.2 Finite Volume Formulation

Drela's finite volume throughflow method is based on the streamline curvature method. The steady state equations, though, are formulated in a finite volume sense.

The first governing equation is the conservation of mass. Across any two adjoining finite volumes

$$\rho_1 v_{m1} \hat{s_1} \cdot \vec{A_1} = \rho_2 v_{m2} \hat{s_2} \cdot \vec{A_2} \tag{2.11}$$

Note that in Equation (2.11) that the $\hat{s_1}$,$\hat{s_2}$ vectors lie along the streamline. Conservation of momentum is written as the steady state Euler equation (Navier Stokes equation *sans* any viscous terms).

$$P_1\vec{A}_1 + (\rho_1 v_{m1} s_1 \cdot \vec{A}_1) v_{m1} \hat{s}_1 + \Pi^- \vec{B}^- = P_2\vec{A}_2 + (\rho_2 v_{m2} s_2 \cdot \vec{A}_2) v_{m2} \hat{s}_2 + \Pi^+ \vec{B}^+ \tag{2.12}$$

Conservation of energy is formulated in terms of enthalpy.

$$\frac{\gamma}{\gamma - 1}\frac{P_1}{\rho_1} + \frac{1}{2}v_{m1}^2 = \frac{\gamma}{\gamma - 1}\frac{P_2}{\rho_2} + \frac{1}{2}v_{m2}^2 \tag{2.13}$$

An additional constraint imposed is that the average pressures on the faces normal to and along the streamtube be equal.

$$P_1 + P_2 = \Pi^+ + \Pi^- \tag{2.14}$$

For the subsonic flow cases which are of interest in marine propeller design, the differential equations form a set of coupled, elliptic, PDE's. The elliptic nature of Equation (2.11) through Equation (2.14) requires an iterative relaxation solution procedure, vice a marching solution technique, which would be used in the case of supersonic flow, where the equations are hyperbolic in nature.

The solution is started by specifying an initial gridline distribution. Each gridline pair is treated as a streamtube. Within each streamtube equations (2.11) through (2.14) are solved simultaneously at each streamwise stations using a Newton-Rhapson technique. The solution produces the pressures Π^+ and Π^- on the streamtube walls. By calculating influence coefficients for the effects of streamline curvature and streamtube area on $\Delta\Pi$, a relaxation equation can be used to update the streamtube positions.

2.2.3 Drela's Throughflow Formulation

Equations (2.11) through (2.14) can be further simplified by forming an explicit scheme that is not differential in nature. The computer code MTFLOW implements such a formulation [3].

The meridional flow speed, v_m is obtained from a local streamtube conservation of mass.

$$v_m = \frac{Q}{\rho A(2\pi r - BT_\theta)} \tag{2.15}$$

The streamwise momentum equation solved within MTFLOW is of the same form as Equation (2.12).

$$dP + \rho v_m dv_m + \rho v_\theta dv_\theta + Pd(\Delta S) - \rho d(\Delta W) = 0 \tag{2.16}$$

The change in work ΔW enters the formulation through the Euler turbine equation.

$$\Delta W = \int \Omega d(r v_\theta) \tag{2.17}$$

If entropy losses or enthalpy additions due to heat release occur within the flow domain, they enter the formulation in Equation (2.16). Equation (2.16) is discretized in a finite volume sense as described in section 2.2.2 to preserve shock capturing.

Instead of solving the differential energy equation, which would require large computational times for iterations, MTFLOW solves for the total enthalpy at any location in the flow.

$$h_o = h_{inl} + \frac{1}{2}v_{inl}^2 + \Delta H + \Delta W \tag{2.18}$$

The final result is a computationally efficient methodology for solving the inviscid equations of fluid motion, which allows for the effects of entropy, enthalpy, and rothalpy to enter the problem.

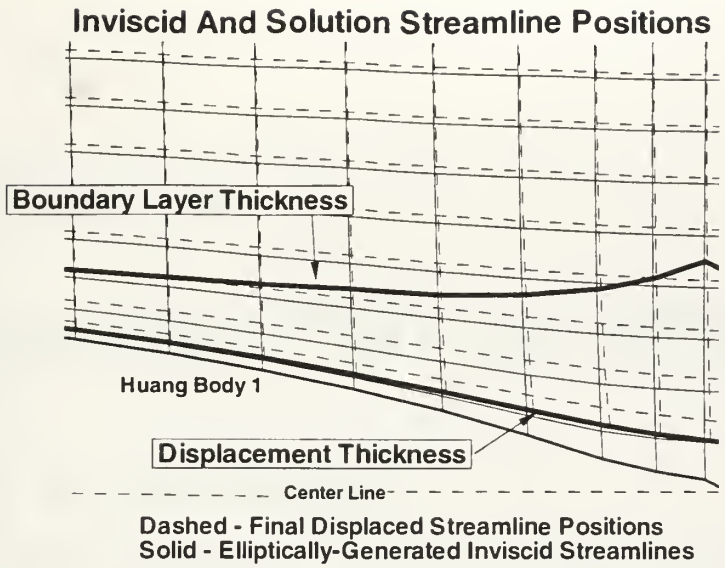


Figure 2-2: The elliptically-generated streamlines and final streamline positions for the displaced body solution for Huang Submarine Body 1. Notice the overlap area between the displacement thickness and boundary layer thickness.

2.3 IBLT Boundary Layer Representation

While the fluid velocity and pressure gradients along the body are known from the inviscid streamline curvature solution, it remains to solve for the flow within the boundary layer. Drela's formulation of the problem uses an integral boundary layer (IBLT) solver to find the boundary integral quantities knowing the inviscid edge velocity and pressure gradients along the solid boundary. Once the boundary layer flow is known, the body boundary is displaced by the boundary layer displacement thickness, and the inviscid outer flow is resolved. In this manner the boundary layer solution and outer potential flow solution are coupled together.

2.3.1 IBLT Diffculties

The difficulty for the propeller designer with the IBLT is that the IBLT solution gives the integral quantities of the boundary layer, such as momentum thickness and displacement thickness. Because the propeller design code requires the total inflow velocity over the entire surface of the blade, and portions of the blade extend through the boundary layer, the velocity profile of the fluid through the boundary layer has to be reconstructed.

The second difficulty encountered is that the boundary layer displacement thickness, δ^*, is less than the actual boundary layer thickness, δ. This is an issue, because the MTFLOW flow solver solution shows high inviscid velocities in the region between δ^* and δ, while in reality, the slower boundary layer fluid velocity exists.

Both of these problems and solutions are highlighted in figure 2-2.

Chapter 3

Coupling PBD and MTFLOW

3.1 Process Overview

The coupling between the streamline curvature code and the propeller blade design code follows that used by Kerwin [14] when he coupled the propeller blade design code with a Reynold's-Averaged Navier-Stokes (RANS) code. The basis of Kerwin's coupling was the use of distributed body forces within the RANS code to represent the presence of the propeller. In the streamline curvature code, the propeller swirl, or angular momentum, is used to represent the presence of the propeller in the flowfield.

In the present coupled analysis or design, one starts by using the propeller blade design code to predict the loading of the propeller based on an assumed flow field. The blade swirl is then transferred to the streamline curvature code where a throughflow solution is computed. To complete the cycle, a total inflow velocity field is extracted from the throughflow domain and returned to the propeller code. The induced velocities predicted by the previous run of the propeller code are subtracted from the total inflow velocity to arrive at a new effective inflow velocity field. This coupling methodology is shown in Figure 3-1.

The coupling is automated through the use of small FORTRAN computer codes. The two computer programs shown in Table 3.1 were written to accomplish the coupling. The actual sequence of running computer programs to run a coupled propeller problem is shown in Figure 3-4.

3.2 Obstacles to Coupling

There were several obstacles to overcome in bringing about the PBD-MTFLOW union.

1. Converting propeller blade force quantities into a measure of energy addition to the surrounding fluid consistent with the streamline curvature basis of MTFLOW.

2. Reconstructing a boundary layer velocity profile based upon integral boundary layer quantities.

PBD2MT	Converts the PBD output circumferential mean induced velocity to swirl. Write out the MTFLO input ascii file containing the swirl, and thickness, if present in the PBD run .
BL2BODY	Converts the MTFLOW output boundary layer data into velocity data (*i.e.*, boundary layer reconstruction), and writes out velocity data field for use by VELCON.

Table 3.1: These two codes are the coupling routines written specifically to couple PBD and MTFLOW. While no changes were made to any PBD code, the MTFLOW code was altered to output TECPLOT-formatted data files to affect the data transfer.

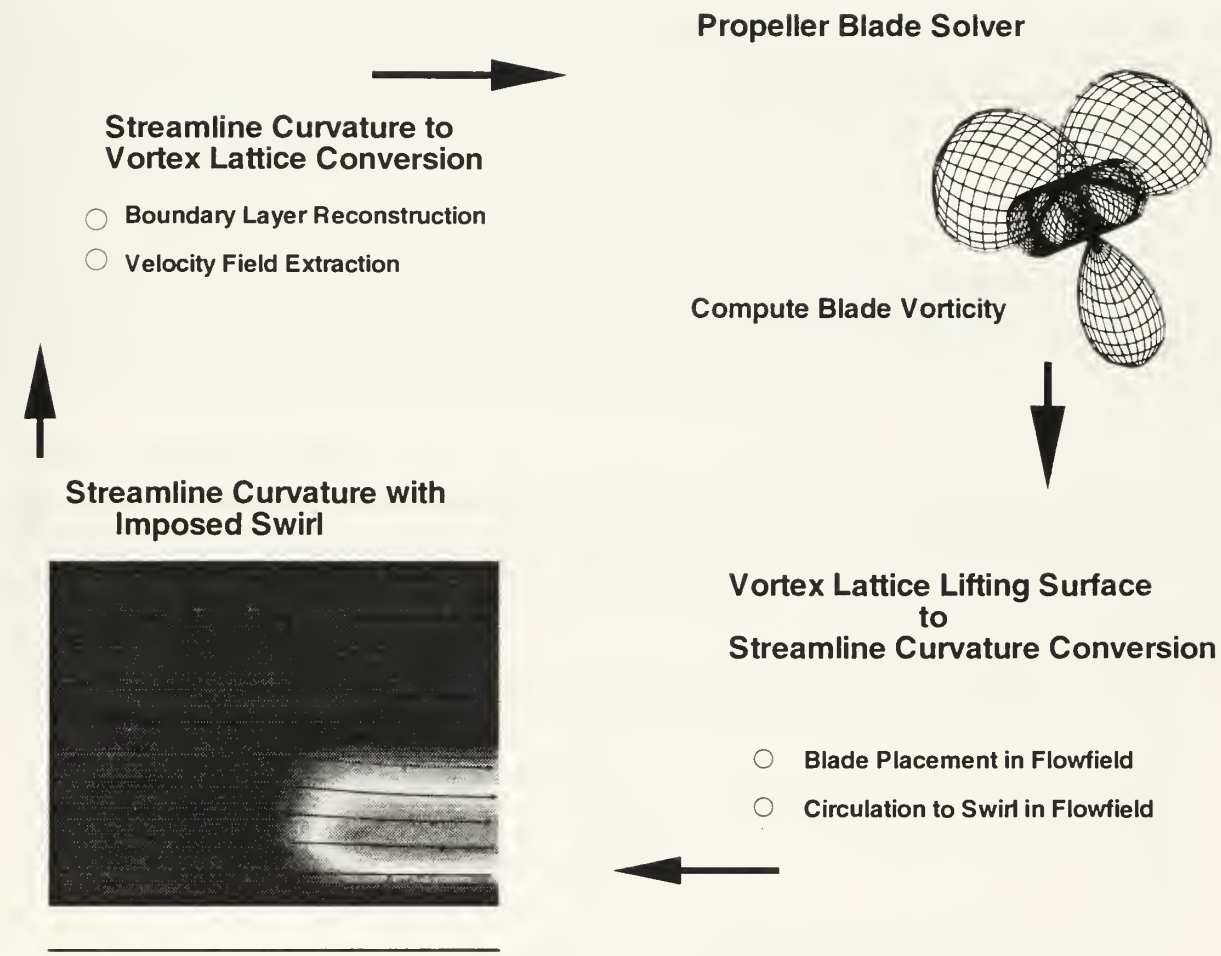


Figure 3-1: The coupling procedure between the propeller blade design code and the throughflow solver.

3. Extracting flow field velocities in a manner such that PBD internal data representation did not fail.

4. Allowing for the severe radial redistribution of streamlines in the case of an open propeller.

The solutions presented in the following chapters represent my proposed solution to the major obstacles. The final chapters, which present validation and results, show that perhaps there is some merit to these solutions.

3.3 Non-Dimensionalization Issues

The important flow quantities passed between the flow solver and PBD are the flow velocities, the propeller rotation rate, Ω, and the propeller swirl, $r\Delta V_\theta$. The flow velocities are used in the blade design code to enforce the kinematic boundary condition on the blade surface. The propeller rotation rate and swirl are used within the throughflow solver to represent the energy imparted to the fluid by the action of the propeller.

3.3.1 Flow Velocities

The magnitude of the velocity is always non-dimensionalized by the inflow speed. This is consistent between PBD and MTFLOW. Therefore, the velocities do not need to be altered when coupling the two codes, even in the case of contracting or expanding, internal, or ducted flows, or in the case of internal flows where the domain inlet velocity is different from 1.0.

3.3.2 Propeller Swirl

The swirl is used in the Euler turbine equation to calculate the work done on the fluid by the rotor. Because of the length scale inherent in the term, the swirl must be scaled by the proper reference length scale when transposing swirl from PBD into MTFLOW.

3.3.3 Propeller Rotation Rate

Internal to PBD, the non-dimensional advance coefficient, J, is always given as follows.

$$J = \frac{V_s}{nD} \tag{3.1}$$

$$J = \frac{1.0 \quad 2\pi}{2R \quad \Omega} \tag{3.2}$$

$$J = \frac{\pi}{\Omega} \tag{3.3}$$

V_s is used to non-dimensionalize the velocities. V_s is usually defined as the speed of advance of the vehicle. In the computation domain where the body is fixed and the flow field has non-zero velocity at infinity, V_s is the value of the upstream velocity at infinity. PBD always assumes that $V_s = 1.0$. It is not correct to use V_A to non-dimensionalize velocities to arrive at the advance coefficient. Notice that the value of the effective inflow velocity in the plane of the propeller swept volume, commonly called the apparent velocity V_A, is not constant throughout the swept volume of the propeller. Hence, the use of V_A would result in ambiguities between experimental and numerical predictions.

It is common to non-dimensionalize empirical propeller tunnel test data with respect to the upstream velocity at infinity, V_s. In internal flow cases, the upstream velocity at the inlet may not be equal to 1.0. Yet PBD assumes that V_s is 1.0. Hence, if V_s is not equal to 1.0, this fact must be reflected by altering the advance coefficient in PBD. This is the so-called effective advance coefficient. The use of an effective advance coefficient requires a change in the usually straightforward calculation of the rotation rate, Ω.

$$J_{\text{eff}} = \frac{1.0\ V_{\text{inlet}}}{nD} \tag{3.4}$$

$$\Omega = \frac{\pi\ V_{\text{inlet}}}{J_{\text{eff}}} \tag{3.5}$$

3.4 Open Propeller Issues

Coupling PBD with MTFLOW to solve for an open propeller presents a challenge because the streamline positions evolve with the solution. To maintain radial equilibrium, the streamlines with high induced mass flow bunch near the centerline unless constrained by a duct. This draws in the lower velocity freestream streamlines to the tip region of the propeller.

The input flow parameter distributions (of, say, swirl $r\Delta V_\theta$, and thickness) are specified with respect to either a fixed (z, r) grid location or a streamtube. Modelling a rotor (or impeller) presents the competing requirements that the swirl input by the rotor does not change physical location, while the shed vorticity (which I also refer to as convected swirl) naturally stays within the same streamtube, which radially contracts and changes position as the streamtube solution evolves. If the shed vorticity does not remain aligned with the local flow, then the wake is not force free.

The problem when using a streamtube solver is that in the case of an open propeller, the streamtubes will constrict in the region of the propeller due to the increased axial velocity of the fluid. This contraction is shown in Figure 3-2. The solution is the use of an elongation factor as explained in Section 3.4.1.

3.4.1 Solutions for Open Propellers

The MTFLO program gives the user the opportunity to specify that input flow changes are fixed in $z - r$ coordinates or in the streamline $s - t$ coordinate system. Fixed in the $s - t$ coordinate system means that the absolute position input flow quantity will evolve as the streamlines evolve with the solution.

In the case of swirl input to the flow domain by a rotor, or removed by a stator, if is not obvious whether the input positions should be fixed or evolve. For while the position of the mechanical blades are fixed, the swirl convected downstream certainly is not.

The solution is to let the swirl evolve with the flow. To counteract the contraction tendency, the radius of the output swirl should be linearly stretched from the hub upward. In other words, with a fixed hub radius, all blade radii are linearly scaled such that the tip radius is set to the elongation factor. Now, as MTFLOW contracts the streamlines towards the hub, the swirl from the blades is moved into the correct geometrical position! The elongation ratio, or stretching factor, is found by a trial and error process. The best way to watch this while running MTFLOW is to guess an elongation ratio, run MTFLOW, and then view the output swirl distribution. The swirl input from the tip of the propeller should be at the geometrically-scaled 1.0 radius value. The proper use of the elongation factor is shown in Figure 3-3.

Since the radial position of the swirl is changed, the swirl input into MTFLOW must be changed, too, such that the strength of the circulation Γ is conserved. Because there is only one length scale involved, it is a linear rescaling.

Notice from Figure 3-2 that not all streamline positions are shifted an equal amount radially. Therefore, a refinement upon the linear rescaling from hub to tip is to alter the radial position of each streamline based on the local streamline shift, vice the global shift. To actually find the shift in streamline position, this thesis implemented a procedure to examine the radial position of the circumferential mean tangential velocity along the trailing edge of the blade output by MTFLOW against the velocity distribution output by PBD. The difference in velocities at any radial positon

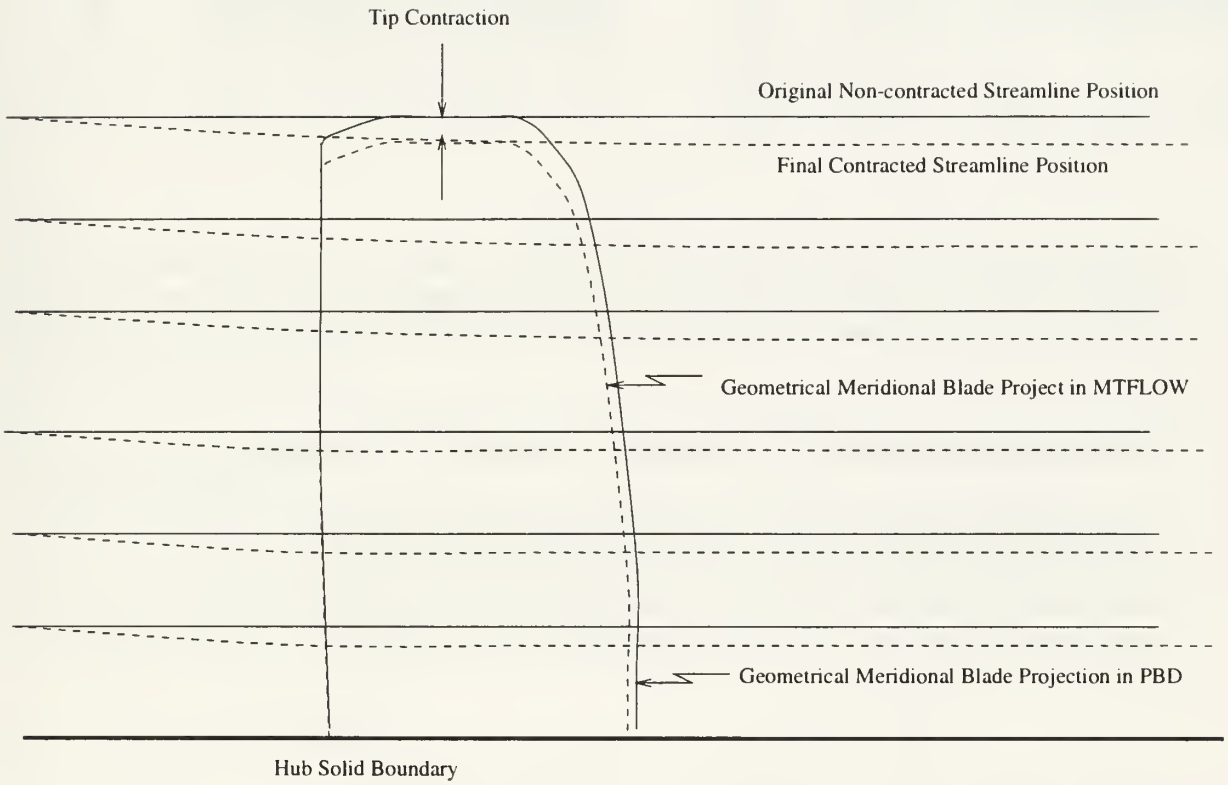


Figure 3-2: Streamline contraction due to the presence of the propeller. Within MTFLOW if the input blade swirl is specified relative to streamline positions, vice physical (z,r) position, the blade swirl is unphysically redistributed with the flow. The the blade design program (PBD) and MTFLOW are solving different flow problems. The surprising this is that the coupling still quickly converges to a non-physical solution.

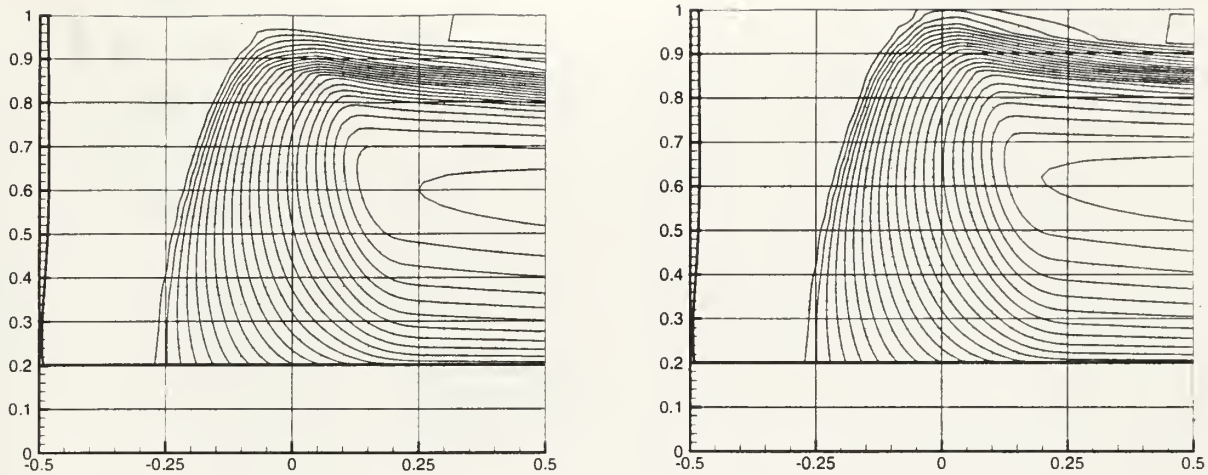


Figure 3-3: .
Use of swirl elongation factor to counter streamtube contraction. Notice that the contour plot of swirl on the left hand picture is contracted such that the tip radius is less than 1.0. By increasing the elongation factor, the right hand side plot shows that the final tip swirl postion is nearly at the physical propeller radius of 1.0

shows the radial redistribution of streamlines, and gives rise to a relaxation scheme to reconverge the streamline positions.

The non-linear radial redistribution scheme is clearly superior for the case of a single blade row propulsor. However, it is not clear that this scheme will still work for multiple blade rows, where the swirl shed by the upstream blade rows must convect, and change the input quantities. While the relaxation scheme would certainly still work (the swirl input into the MTFLOW domain should be redistributed until it is not changed by the MTFLOW solution), it is not clear that convergence can be obtained as quickly, if at all.

3.4.2 Swirl Convection

The user is required to manually convect trailing swirl from the trailing edge of the blade row to the domain exit. While this is a well-defined problem for internal flow passages, where the streamline positions can not radically change, the open propeller (*i.e.*, external propulsor) presents unique challenges. From basic empirical observation as well as numerical simulation, it is known that the wake of an open propeller contracts. To maintain a force free wake, the swirl convected within the wake must contract with the streamlines.

Several coupling methods were tried to overcome the difficulties with wake convection.

1. Convect trailing wake with no radial contraction. This solution is the easiest to implement however it results in the physically unrealistic flow situation that the wake is no longer force free. This line of thought was not pursued to investigate the magnitude of the problem.

2. Convect Wake Along previous iteration's streamlines. MTFLOW is used to solve the flow solution with the convected swirl fixed in (z, r). The convected swirl is placed downstream based on the previous iterations streamline positions. After the MTFLOW solution converges, the wake is re-aligned based on the updated streamlines, and the solution is re-converged. This wake re-alignment procedure is repeated two times before a new propeller blade solution is computed.

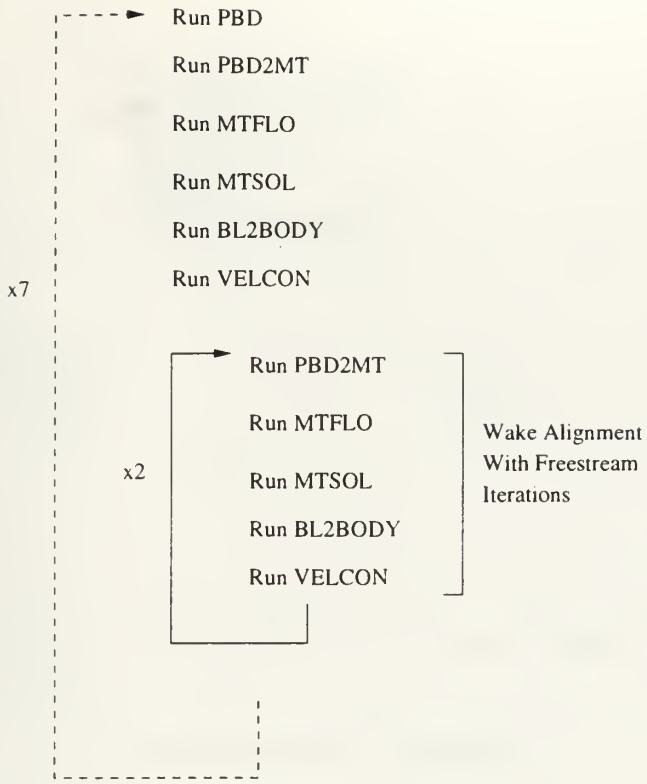


Figure 3-4: The coupling procedure between the propeller blade design code and the throughflow solver for an open propellor case. The inner iteration loop is required to align the trailing vorticity with the freestream velocities.

The final method works the best from a computational efficiency and accuracy standpoint. The iteration cycles are illustrated in Figure 3-4.

In practice, after the initial MTFLOW solution has converged, the wake streamlines change very little, and it is practical to skip the inner wake alignment loop shown in Figure 3-4.

3.4.3 Swirl Radial Redistribution

Circulation shed off the propulsor blades is represented as swirl convecting along streamlines in MTFLOW. As the radius of the streamline about the machine's centerline changes, the strength of the swirl, $r\Delta V_\theta$, must change to maintain constant circulation, Γ. There is a direct inverse relationship.

$$\Gamma \propto f\left(rV_\theta\right)$$

This is an important consideration when convecting swirl through the meridional passage of a mixed flow machine and along the wake of a free propeller. If the swirl strength in the MTFLOW input were not altered to maintain constant circulation, the wake in MTFLOW would appear to have windmills removing energy in it as the radial position was altered. Recall that there can be no radial streamline redistribution due to wake vorticity, because there is no poewr being put into the fluid, as evidenced by the fact that Ω equals zero in the trailing wake system.

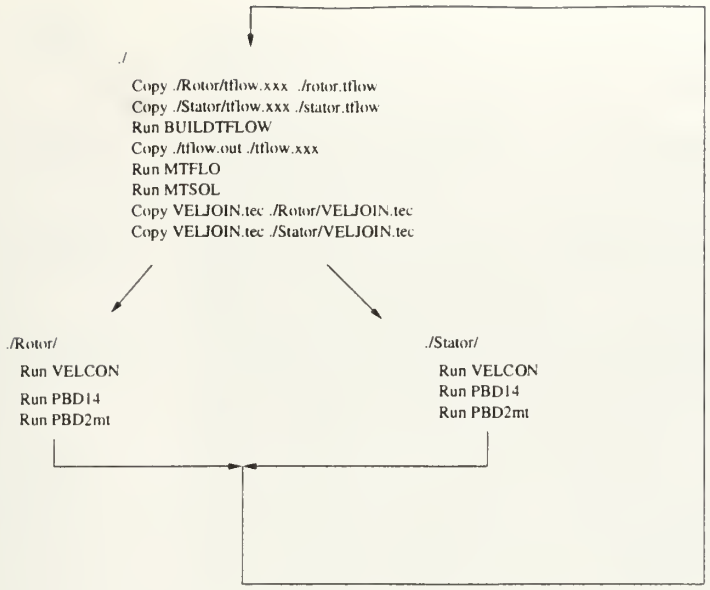


Figure 3-5: Multiple Blade Row Coupling Flowchart. This process is easily automated in a script file.

3.5 Running the Coupling

PBD2MT automatically assumes that the blade row described in the PBD output files is the only blade row that exists in the throughflow domain described in the MTFLOW walls.XXX file. If more than one blade row is present, PBD2MT must be run on each blade row, and then the BUILDTFLOW program must be executed. This process is shown in figure 3-5.

3.5.1 Coupling Admin File:MTCOUPLE.INP

The mtcouple.inp file is the only file which the user must create from scratch. It serves as an administration file.

```
LINE 1: 3                          ! Reynolds number
LINE 2: 0.00955                    ! inlet mach number
LINE 3: 0.4211                     ! Vship used in PBD to calculate Js
LINE 4: 9                          ! aft most point of body
LINE 5: 0.2                        ! x location of LE tip
LINE 6: 1.0                        ! r location of LE tip
LINE 7: tflow.rotor                ! tflow file name
LINE 8: walls.rotor                ! walls file name
LINE 9: rotor.pbd                  ! pbd14.4 input file name
LINE 10: 1.000                     ! Elongation factor
```

The input lines have the following uses within the code:

LINE 1 : REYNOLD'S NUMBER. Reynold's number for the throughflow domain. If less than 10, BL2BODY code assumes an inviscid flow domain and no boundary layer reconstruction takes place.

LINE 2 : INLET MACH NUMBER. For incompressible flows, the mach number is defined as the velocity divided by the speed of sound. For water, $M = \frac{q}{1660\frac{m}{s}}$. See section A.5 for further details.

LINE 3 : VSHIP. This is the upstream velocity at infinity. For coupling purposes it is the fluid velocity at the domain inlet. Remember that the advance coefficient, J, within PBD is calculated using Vship.

LINE 4 : AFT MOST BODY POINT. Only used in the case of an external flow domain where the stern of the body terminates in the throughflow domain. Downstream of this point, BL2BODY will use its own wake model to reconstruct the boundary layer velocity profiles vice Swafford relations.

LINE 5 : X-LOCATION OF LE TIP. Given in throughflow coordinates, the axial location of the propeller leading edge tip – similar meaning as in VELCON.

LINE 6 : R-LOCATION OF LE TIP. Given in throughflow coordinates, the radial location of the propeller leading edge tip – similary meaning as in VELCON.

LINE 7 : TFLOW FILE NAME. Name of the MTFLOW tflow data file which will be output by PBD2MT.

LINE 8 : WALLS FILE NAME. Name of the MTFLOW walls geometry data file.

LINE 9 : PBD14.4 INPUT FILE NAME. Name of the PBD14 admin file. Used to read the advance coefficient and number of blades.

LINE 10 : Elongation factor as explained in Section 3.4.1.

Chapter 4

PBD to MTFLOW Conversions

4.1 Program Overview

The coupling between PBD and MTFLOW occurs using the PBD2MT program. The file passing and interaction between PBD and PBD2MT is shown in Figure 4-1. The goal of the PBD2MT program is to produce an MTFLOW input ascii file, referred to as a *tflow* file in MTFLOW parlance, that contains a description of the swirl, entropy, thickness, and enthalpy input into the flow field by the propeller blades. Currently, PBD2MT handles swirl and thickness only. Entropy generation from the viscous drag on the blades could be added if desired, and enthalpy is not applicable for a propeller if the propeller forces are transferred via swirl.

PBD2MT converts the PBD14-calculated circumferential mean propeller induced velocities to swirl, rV_θ . The program also automatically checks for blade thickness and writes the blade thickness to the output file, as well. The user is only required to make a single administration input file to run the code. It would be a relatively simple matter to input the entropy generated by the blades based on the blade sectional drag into the meridional flow using the current framework.

4.2 PBD2MT Input/Output Files

The only specific input file the user must create is the coupling administrative file, called mtcouple.inp. The rest of the files are either created by the successful completion of a Mode 6 PBD14.4 run, or files required by MTFLOW. All of these files should be placed in the same directory where the PBD14 output is written to ensure data is not lost or inadvertently overwritten.

mtcouple.inp	administrative file for PBD2MT program Lists flow paramters and input file names
XXX.pbd	PBD14.4 style input file Lists the Advance Coefficient, J_S
PBDOUT.CMF	output file from blfrc.f with grid and control point (CP) positions as well as azimuthal blade thickness at CPs
PBDOUT.CMV	circumferential mean induced velocities at CPs
WALLS.xxx	MTSET style walls file
restart.vel	flowfield velocity description in tecplot format used for wake trajectory alignment

Table 4.1: PBD2MT required input files

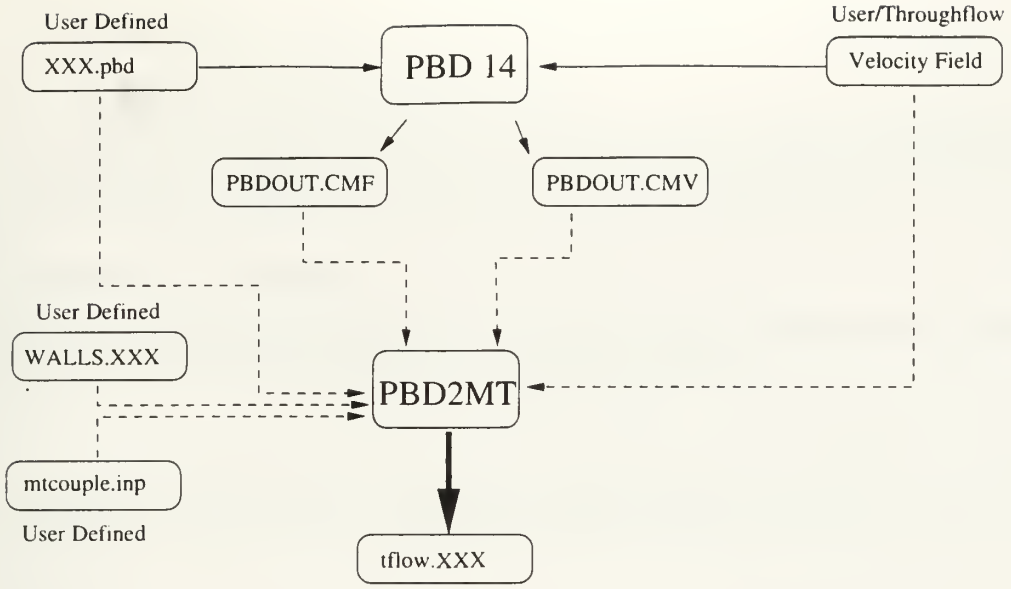


Figure 4-1: The interaction and file passing between PBD and PBD2MT. Notice that four external files are required to run the coupling: the PBD input file, the MTFLOW WALLS.XXX file, the flowfield file (usually called restart.vel), and the PBD2MT admin file, mtcouple.inp. The end product is the ascii MTFLOW input file, TFLOW.XXX

tflow.XXX	TFLOW file containing swirl and thickness distribution in the flow domain. Inputs directly into mtflo.
p2mout.sgr	The following output file is commented out in the code but could be used for consistency checks. Calculation for blade circulation at the trailing edge in same format as PBDOUT.SGR.

Table 4.2: PBD2MT Output Files

4.3 Program Flow

Three main tasks are required to convert PBD-14 output data to a form suitable for MTFLOW use.

Task 1 read the walls.XXX file with the body geometry
Task 2 Rescale the propeller geometry to the throughflow geometry scale
Task 3 write a tflow.XXX file with the prescribed swirl, thickness, and entropy from the PBD output

4.3.1 Task 1 : Read walls.xxx

The walls.xxx file contains the data points of the solid boundaries. This same file is used by MTSET. PBD2MT reads the inlet and outlet x locations, and the solid boundary points. Since allocatable arrays are used in PBD2MT, the maximum number of points on any element is set by the variable MAXSIZE in PBD2MT. If it is exceeded an error message is produced.

For increased accuracy, the user should place many points along the hub and duct in the area of the blade, because PBD2MT uses linear interpolation to avoid spline interpolation overshoots. Specifically, a boundary point should be placed immediately upstream and downstream of the blade hub and tip to ensure a good interpolation. Also, the user should well-define any regions of large

curvature gradients in the hub and tip region of the domain.

4.3.2 Task 2 : Geometry and Swirl

The TFLOW.XXX file involves specifying the swirl $r(\Delta V_\theta)$, thickness, and entropy distribution throughout the fluid domain. PBD2MT computes these quantites at the PBD blade control points, as explained below. The heart of this process is the aoutput.f subroutine.

Blade Row Rotation Rate

PBD2MT reads the PBD Administration file to get the advance coefficient and the number of blades. The blade row rotation rate is computed from Equation (4.1).

$$\Omega = \frac{\pi V s}{J_{eff}} \tag{4.1}$$

As explained in Section 3.3.2 V_S may not equal 1.0. Hence, the value for V_S must be included in the mtcouple admin file. If $J_{eff} \geq 10.0$ then PBD2MT assumes a stator.

Geometry

PBD2MT reads the PBDOUT.CMV and PBDOUT.CMF files to get the following data:

1. X,Y,Z location of lattice grid
2. X,Y,Z location of control points
3. (from .CMF) Azimuthal blade thickness at each control point
4. (from .CMV) u, v, w circumferential mean induced velocity components at the control points

Circulation to Swirl Conversion Equations

The governing equation for the conversion of circulation to tangential swirl velocity is repeated here for clarity.

$$\vec{\Gamma}_B(r) = \frac{-2\pi r V_\theta^*}{Z} \tag{4.2}$$

Reversing the terms in Equation (4.2) gives the relationship for the tangential swirl in terms of circulation.

$$r V_\theta^* = \frac{\vec{\Gamma}_B(r) Z}{2\pi} \tag{4.3}$$

Equation (4.2) and Equation (4.3) are both required in the conversion process.

Circulation to Swirl Implementation

There are five basic steps to convert circulation to swirl.

1. PBD outputs the circumferential mean induced velocity u, v, w components at every control point on the blade in the PBDOUT.CMV file. A transformation to cylindrical coordinates yields the circumferential mean induced tangential velocity component. Now, the circulation at every control point is known through equation (4.2).
2. The circulation Γ is scaled by the reference length common between PBD and the throughflow domain. This reference length is always the propeller radius in throughflow coordinates, because in PBD this length is always 1.0. Circulation Γ does not usually need to be scaled by a reference velocity, since in both PBD and MTFLOW, all velocities are non-dimensionalized by the inlet velocity.

3. In the case of an open propeller the swirl may have to be elongated, as explained in Section 3.4.1. To maintain constant Γ, if the radius is increased, the swirl rV_θ must be decreased.

4. After non-dimensionalizing all arrays containing a length scale, the swirl, array name RVT in the source code, is then re-calculated using Equation (4.3). Finally, because MTFLOW is a compressible flow code, the isentropic compressibility factor must be applied to the swirl.

$$r\left(\Delta V_\theta\right) = \mathrm{RVT}\left(1.0 + 0.2M^2\right) \tag{4.4}$$

4.3.3 Task 3: Output to the tflow file

The tflow.XXX file is broken down into blocks of data, which correspond roughly to span lines of constant percent chord length along the blade. In PBD14-speak, these are lines of constant n. Whenever there is an abrupt change in one of the specified flow variables, such as at the leading edge or trailing edge of the blade, a spline break is introduced by doubling that span line.

One implementation key is that MTFLOW can only accept a set number of t values. The radii of the control points in PBD are determined by the local streamline vice a constant distance off of the centerline of the machine. The distributions must be output along constant spaced t coordinates. This requirement requires a cubic interpolation of the swirl from the PBD output radial position to the constant t coordinate.

4.4 Coversion to MTFLOW (s,t) Coordinate System

One key to the coupling process is the rescaling of the propeller swirl such that in MTFLOW $s-t$ coordinates, the radial t coordinate goes from 0.0 at the hub to 1.0 at the edge of the domain. This means that if the user writes out a r coordinate in the tflow.XXX file, r, too, must go from 0.0 to its ultimate value. In general, the relationship between t and r is shown in Equation (4.5).

$$t_i = \left(\frac{r_i}{R_{max}}\right)^2 \tag{4.5}$$

The local streamline radius, r_i, is calculated by first dividing the flow channel between the hub and tip into equally spaced subdivisions, and then finding the non-dimensional distance of each subdivision from the hub.

$$\Delta r = \frac{R_{tip} - R_{hub}}{NI}$$

where NI is number of radial positions

$$r_i = \frac{i\Delta r}{R_{tip} - R_{hub}}$$

where i is local radii index

4.5 Important Usage Tips

4.5.1 walls.xxx File

If an axisymmetric element extends to the upstream and downstream extent of the fluid domain ,such as in the case of the inlet and outlet of a waterjet, extend the geometry definition of the hub and casing slightly ahead of the XINL specified in the walls file. This will keep PBD2MT from having problems of finding the inlet radius. If this is not done, PBD2MT will write out a very strange *tflow* file.

4.6 Notes on the code PBD2MT

4.6.1 Main program: pbd2mtv2.f

1. Read the *mtcouple.inp* file.
2. Read the WALLSNAME file. Converts the counterclockwise ordered data points into inlet-to-outlet ordered boundary points.
3. Read the pbd admin file to get number of blades and rotation rate.
4. Read PBDOUT.CMF to get blade locations, CP location, azimuthal thickness.
5. Read PBDOUT.CMV to get the TANGENTIAL induced swirl velocity.
6. Read restart.vel to get streamline information.
7. Rescale streamlines by SFAC.
8. Convert length scales from PBD to MTFLOW scale.
9. Call subroutine *aoutput* to write out tflow file.

4.6.2 Subroutine aoutput.f

This subroutine bears the responsibility for writing out the ascii tflow.XXX file for use in MTFLO. It has several components.

1. Calculate the constant t and r values where data will be output.
2. Write out the header lines of the tflow file.
3. Write out a bunch of zeroes at the inlet.
4. Write out zeroes at LE of blade.
5. Double LE of blade and write out correct rotation rate.
6. Loop down every blade span line and write out by
   (a) Spline x,rv_θ, and thickness with respect to r.
   (b) At every output r radius, interpolate the splines and write out values of x,rv_θ, and thickness.
7. Double the specified value at the blade TE.
8. Write out user specified number of convected wake stations between blade TE and domain outlet.

Chapter 5

MTFLOW to PBD Conversions

The main challenges in converting MTFLOW output back to PBD input was the reconstruction of the boundary layer velocity profiles, and matching these boundary layer profiles with the displaced body inviscid velocity flow field found by MTFLOW. The flow of input and output files is shown in Figure 5-1.

5.1 BL2BODY Guide

The following two sections present how to run BL2BODY and then theory behind the actions performed by BL2BODY.

5.1.1 Running BL2BODY

The input files required by BL2BODY are shown in Table 5.1. The output files created by BL2BODY are shown in Table 5.2. All files are in TECPLOT format.

5.2 Velocities from MTFLOW to PBD

PBD expects to receive the nominal wake from the flow solver. Internal logic within PBD extracts the effective wake. This requires an accurate representation of the velocity field in the region of the propeller, and downstream of the propeller. Therefore, BL2BODY must reconstruct a boundary layer velocity field through the IBLT displaced body region and attach it with the inviscid solution.

5.3 Boundary Layer Reconstruction

The Integral Boundary Layer solver within MTFLOW solves for the gross boundary layer quantities such as shape factor H, momentum thickness θ, displacement thickness, δ^*, edge velocity, and shear

1. mtcouple.inp	Coupling administrative file. Fully described in section 3.5.1.
1. OUTBL.tec	Boundary Layer Quantities from MTFLOW: XSSI, H, DHSDHK, THETA, DSTR, UEDG, CTAU, TAUW, DISPq. Output by MTSOL.
2. ORIGGRID.tec	Original elliptic grid set by MTSET. Output by MTSET.
3. OUTVEL.tec	Outer, inviscid, velocities from MTSOL solution. Output by MTSOL.

Table 5.1: BL2BODY required input files.

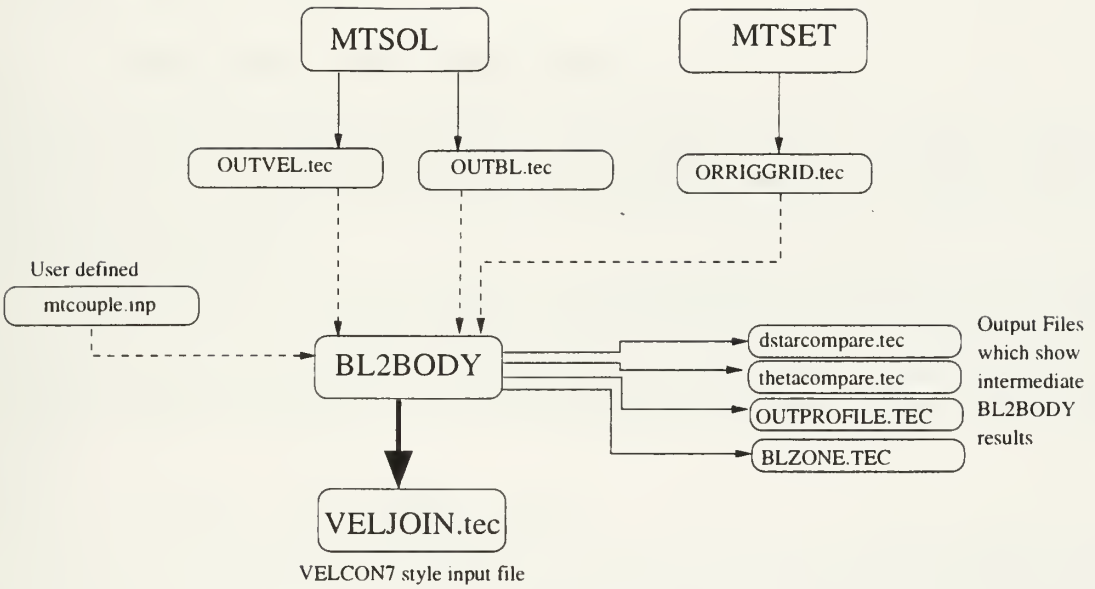


Figure 5-1: The interaction and file passing betweenMTFLOW and BL2BODY. The end product is the ascii, TECPLOT format file VELJOIN.tec with the velocity field. This file is formatted as a VELCON7 input file.

OUTPROFILE.TEC	Boundary layer velocity profile as separate zones
BLZONE.TEC	Boundary layer velocity profiles as one large zone Makes for easier viewing of contours, etc ...
VELJOIN.TEC	Final flow field velocity with boundary layer and and inviscid flow joined together. Input file for VELCON71.
	Debugging Analysis files
dstarcompare.tec	MTFLOW displacment thickness δ^* along the body and BL2BODY final output δ^*
thetacompare.tec	MTFLOW momentum thickness θ along the body and BL2BODY final output θ

Table 5.2: BL2BODY Output Files.

stress coefficient. Because often times the root of the propeller blade extends through the boundary layer, it is necessary to recreate the fluid velocity profiles through the boundary layer. Indeed, in the full stern submarine concept, the entire propulsor is within the boundary layer. At the heart of the boundary layer relations is the closure relation. The current implementation uses the Swafford correlations as closure relations.

5.3.1 Swafford Boundary Layer Profile

Swafford presents an empirical correlation method relating the fluid velocity ratio $\frac{u}{u_e}$ across the boundary layer to the profile shape factor, H, and momentum thickness Reynold's number based on edge velocity Re_θ [23]. The methodology for genarating boundary layer velocity profiles is well documented in the previously cited reference. These velocity profiles are valid for attached and separated turbulent flows.

The velocity profile within the boundary layer is first computed in local body normal ξ-η coordinates using Swafford's correlation. Values for the local shape factor, H, and Re_θ are provided from the MTFLOW solution. Swafford's procedure yields the total velocity magnitude. This velocity profile is checked to ensure that it matches the displacement thicknes, δ^*, output by the MTFLOW solution. If it does not, the profile shape is linearly adjusted until the displacement thickness matches that of the MTFLOW solution.

5.3.2 Boundary Layer Wake Model

Analytically resolving the boundary layer velocity profile in the wake near the trailing edge of a body has been a great source of difficulty for those researchers not pursuing a volume representation of the fluid domain. A similarity solution to the wake profile is valid in the laminar case only after three body, or chord, lengths downstream of the body. This fact leads most researchers to deduce an interpolation scheme from the body boundary layer, through the near wake area, to the far wake (3 body lengths downstream) where analytically-derived, empirical relations exist.

For this research, the analytical approach derived by White was followed [25]. White shows that the velocity defect within a laminar axisymmetric wake is given by:

$$\frac{u_1}{U_O} = C_D \frac{U_O L}{8\pi\nu} \frac{L}{x} \exp(-\frac{U_O r^2}{4x\nu}) \tag{5.1}$$

If equation (5.1) is applied to a turbulent wake, velocity ratios greater than 1.0 result. To avoid this situation, this research modified equation (5.1) such that a Gaussian distribution of velocity across the wake was maintained:

$$\frac{u_1}{U_O} = (1.0) \frac{X_{TE}}{X_{far}} \exp(-\frac{r^2}{X_{far}^2 \alpha}) \tag{5.2}$$

where	X_{TE}	X location of body TE
	X_{far}	X location where far wake approx. valid
	r	local radius
	α	adjustment factor

The adjustment factor α is adjusted such that the momentum thickness of the resulting profile shape is equal to the momentum thickness given by the MTFLOW soltuion.

5.3.3 Fitting Boundary Layer Velocity Field to Flow Field

Using either the straight Swafford relations or the modified wake relations, a velocity is profile is found through the boundary layer. It now remains to body fit the ξ-η coordinates to the body boundary. Since the boundary layer velocity profile is found in a normal coordinate system, it would seem natural to merely rotate the ξ-η coordinate system such that the η normal direction vector matched the vector normal to the body. This method is incorrect, however, because the boundary layer velocity vectors do not conform to the flow streamlines. This method also suffers from the shortcoming that along concave body surfaces, the boundary layer profiles will intersect at a distance equal to the local radius of curvature off the body.

A second method is to keep the velocity vectors normal to the body, but lay the velocity vectors along a gridline semi-normal to the body. This method prevents boundary layer velocity profiles from crossing, but does not ensure that the boundary layer velocity vectors follow streamlines.

A good compromise, then, is to lay the velocity vectors along an elliptically-generated gridline semi-normal to the body, and force the velocity vectors of the boundary layer flow to exactly conform to the inviscid streamlines. Near the body, where the inviscid streamlines are displaced off the body by the displacement thickness, the streamlines from the original elliptically-generated grid are used – the reader will recall that the grid points in an elliptically generated grid numerically satisfy Laplace's equation, and, thus, are inviscid streamlines. This compromise method gives very good results.

5.3.4 Joining Boundary Layer Velocity Vectors and Outer, Inviscid Velocity Vectors

MTFLOW solves for the inviscid fluid velocity outside of the displacement thickness, δ^*. δ^* is always less than the boundary layer thickness. There is an overlap region , then, between the boundary layer velocity profile, which extends from the body to a distance δ off the body, and the solved, inviscid velocity flowfield, which extends from a distance δ^* off the body to the edge of the fluid domain. The velocity within the overlap region was taken as the boundary layer velocity for 90 percent of the boundary layer region – arbitrary, but reasonable. To correct for any interpolation problems, the solved-for boundary layer velocity is adjusted by a constant factor such that the velocity at the edge of the boundary layer matches the inviscid velocity within a streamline near that point. Since this linear scaling alters the displacement thickness and momentum thickness of the boundary layer, the boundary layers must be adjusted to match the outer inviscid velocity and the momentum thickness. The final δ^* of the velocity profile closely matches the MTFLOW-derived δ^*.

Swirl in the Boundary Layer

Because the tangential swirl velocity is not carried in the boundary layer quantities, the inviscid swirl must be smeared across the boundary layer to give good results. If the swirl is not represented correctly, the blade section within the boundary layer will have the wrong pitch angle.

The tangential velocity is interpolated across the boundary layer. The tangential velocity along the boundary is considered 50 % of the tangential velocity at the inviscid edge of the boundary layer. In between a linear interploation is done to transition for the value at the wall, to the value at the inviscid edge

5.4 Notes on the code BL2BODY

The code is written in FORTRAN 90. Where possible, allocatable arrays are used and destroyed within individual subroutines. IMPLICIT NONE is used to ease debugging. All data is passed between subroutines as arguments. There are no common arrays. Where parameters are hard-wired into the code, the offending subroutine header is annotated, as well as comments placed in areas of the code affected.

5.4.1 Main Program : bl2body.f

The file *bl2body.f* controls the sequence of events required to convert MTFLOW velocity and boundary layer data into a flowfield suitable for use by PBD-14.

1. Read the administration file, *mtcouple.inp*
2. If the Reynold's number is less than 10.0 *BL2BODY* assumes that this is an inviscid flow case and calls *outvel2veljoin*.
3. Read the boundary layer data file called *OUTBL.tec* which is output by the modified MTFLOW *io.f* subroutine.

5.4.2 Subroutine sprof.f

This subroutine computes the velocity at any point within the boundary layer based on the momentum thickness-based Reynold's number Re_θ and boundary layer shape factor H. The routine is attributable to Swafford [23]. If the routine has any problems matching the correlations, usually due to trying to evaluate the velocity in a region of laminar flow, the subroutine automatically switches over to a $\frac{1}{7}^{th}$ Power Law velocity profile calculation.

5.4.3 Subroutine joiner.f

The boundary layer velocity profile data previously found is passed in. Because the boundary layer data is already in the correct x, y position it now must be joined with the inviscid velocity solution from MTFLOW. The inviscid solution is written out by the modified MTFLOW subroutine *io.f* in the file OUTVEL.tec.

The steps accomplished in *joiner.f* :

1. Read in the OUTVEL.tec invsicid velocity data.
2. If there is fluid along the centerline ahead of the axisymmetric, centerline body (*e.g.*, in the case of a shaft with a propeller on it which extends from the downstream portion of the flow domain into the flow domain) interpolate the inviscid velocities and write to the output file.
3. Shift the boundary layer velocity data such that the velocity is given at the cell centers, vice nodes
4. For each vertical line of boundary layer data
   (a) Find the inviscid velocity point closest to the end of this line
   (b) Consider this point the inviscid edge velocity
   (c) Linearly alter the boundary layer profile previously found to match this inviscid edge velocity
5. Calculate the new boundary layer displacement and momentum thickness for the user to visually compare with the MTFLOW solution for the displacement and momentum thicknesses.

5.4.4 Subroutine velconout.f

The only purpose of this subroutine is to take the joined velocity file written out by *joiner.f* which is j-ordered data, and turn it into i-ordered data. This subroutine:

1. Open and read VELJOIN.tec
2. Over-write VELJOIN.tec in i-ordered data format

Chapter 6

Validation

Validation of the MTFLOW-PBD coupling was carried out in several stages, following the "build a little, test a little" philosophy. These tests built from the testing of individual components, and progressed through the validation against empiricial test cases.

Because the two underlying computer codes are well-validated, it was not necassary to verify the internal accuracy of the codes. Rather, logical consistency between the codes was required. This was done by using simple propeller models, such as one with zero thickness, inviscid blades, and showing consistency between inviscid theory and MTFLOW.

The coupling technique was validated in several stages. Starting from validation of the basic flow solver abilities, each successive level of validation represents an added level of complexity. By the last validation, a fully viscous internal flow case is tested. Table 6.1 shows the validation tests conducted.

6.1 Validation: Nominal Wake

As previously mentioned, if there is no vorticity present in the inflow (*i.e.*, an inviscid inflow with no radial velocity gradients), there is no vorticity to interact with the propeller, and no inflow vorticity to redistribute. Hence, the effective inflow must equal the nominal inflow. This fact is used in validating the coupling between the propeller design code and the throughflow solver to ensure that the same forces, or power in this case, are modelled within the throughflow solver and the propeller code.

Modelling a propeller on an infinitely long hub of constant radius removes any hub and duct effects from the circumferential meridional flow. Thus, we can assume a fluid inflow velocity which

Open Propeller, inviscid	To validate the ability of the two codes to couple and converge the nominal and effective wakes.
No propeller, viscid	To validate the ability of MTFLOW to accurately predict low mach number, high Reynold's number flow situations.
Open Propeller, viscid	To validate the abilityof the two codes to couple and converge to experimental results
Ducted propulsor, inviscid	To validate the ability of the two codes to recreate empirical ducted propulsor results.
Ducted propulsor, viscid	To validate the boundary layer routines and coupling technique.

Table 6.1: Validation of the coupling technique involved increasing levels of fidelity and complexity.

is constant across all radii. This is the nominal inflow. After the PBD-MTFLOW coupling has converged (usually after about 6 interations), we can extract the effective velocity from the total flowfield. If this effective velocity from the coupled solution matches the nominal velocity of 1.0, we will have shown that the codes couple to the correct answer.

The propeller code solves for the blade circulation given the constant inflow. The circulation is input into the throughflow solver as swirl. The throughflow solver solves the equations of fluid motion. In so doing, the fluid velocity is increased due to the presence of the propeller, streamtube contraction occurs across all propeller radii, and wake contraction becomes evident downstream of the propeller. Once the flow solution is converged, the meridional velocity field in the region of the propeller is extracted, and converted for input into the propeller design code. Those are the steps which comprise one coupling iteration. Now, the propeller code re-solves for the blade circulation distribution with this new inflow. This process takes approximately six coupling iterations to produce converged results. Results are considered converged when the circulation distribution on the propeller does not change with successive throughflow coupling iterations. Conversely, the throughflow flowfield velocity solution does not change with successive couplings.

6.1.1 KA 455 Rotor

A Kaplan KA 455 rotor was used for this validation. The nearly radial projected blade shape reduced problems with MTFLOW splining. Figure 6-1 shows the results for a Kaplan Series 4-55 rotor with no duct placed in a nominal inflow of unit velocity. Notice the over the majority of the blade, the effective velocity is with 1 percent of the nominal velocity. The missed regions in the hub indicate that there may still be some interpolation problems which need to be smoothed out near the hub. This coupling was performed using the non-linear streamline radial redistribution methodology.

6.1.2 Nominal And Effective Calculation on the Fly

Since the nominal and effective inflow velocity fields must be equal for the propeller in an inviscid flow, there is another sure-fire method to test any new coupling routine. This is to compare the radial load distribution predicted by the coupled codes with the radial load distribution predicted with the propeller in the nominal inflow field, with the caveat that the wake trajectories from the current meridional flow solution are used to ensure that wake convection between the two calculations is exactly the same. If the solution is not converging rapidly this method very quickly eliminates the wake sheet geometries as the culprit, since the same wake sheets are used, now, in both calculation.

One key is to get the same trailing wake sheet geometries between the effective calculation and the coupled calculation. This involves specifying the local wake convection axial and tangential induced velocity components in the PBD Mode 4 input file[16]. The radial contraction portion is accounted for by the fact that PBD evolves the wake along the input streamlines. To find the induced velocity component in the wake, the circumferential mean solution from the throughflow solver is used. Since the upstream flowfield is uniform with strength 1.0, any difference in velocity at any point from 1.0 must be due to an induced velocity component. Now that the circumferential induced velocity is known in the wake, the wake is thus aligned correctly.

As a further refinement, the position of the streamline positions from the throughflow solution are also used. That is, the streamline positions from the VELCON output file are used. The velocity along these streamline is set to axial, uniform inflow. Now the user is assured that the vortex lattice geometry used in the coupled and uncouple, stand-alone effective wake calculation are the same.

Once the radial circulation distribution is found from the uniform effective flowfield, the coupling is run. When the solution converges, since the nominal wake must equal the effective wake the propeller code should give the exact same blade loadings. The results for this test, again performed with a Kaplan KA 4-55 rotor, *sans* duct, are shown in Figure 6-2. The results from this analysis show a completely converged solution.

In performing this validation, the required radial redistribution of swirl problem was highlighted. It became apparent that the linear redistribution of radii would not produce the correct results. Therefore, the radial position of each constant span line of swirl input into MTFLOW was altered by

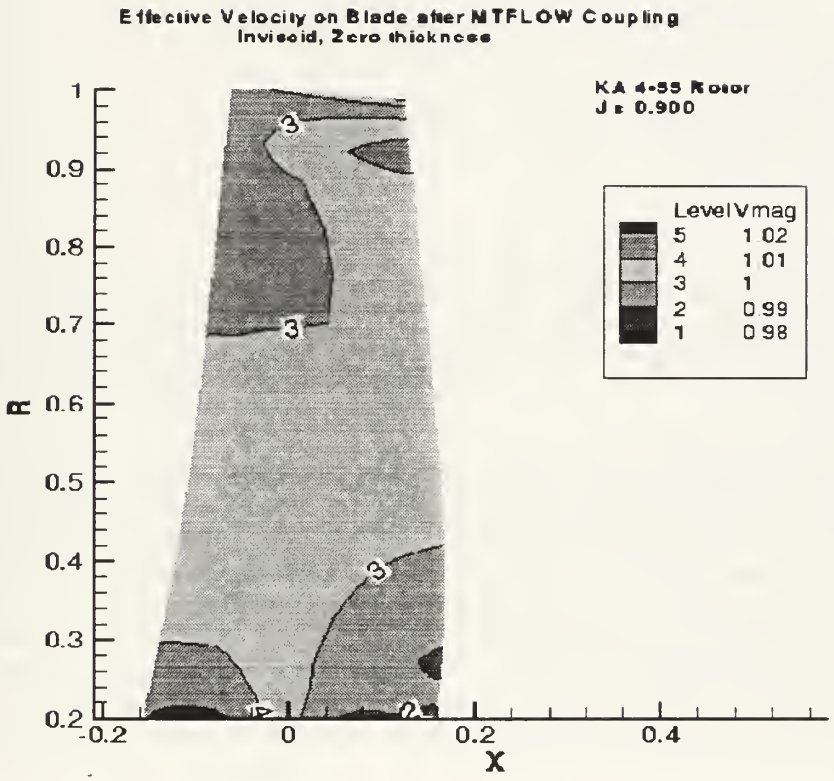


Figure 6-1: .

Given a nominal axial inflow velocity of 1.0, the axial effective inflow velocity shown in the contour plot should equal 1.0 along the entire blade. The small portions of the blade not equal to 1.0 , level 3 on the chart, are due to inaccuracies in the present streamline adjustment methodology.

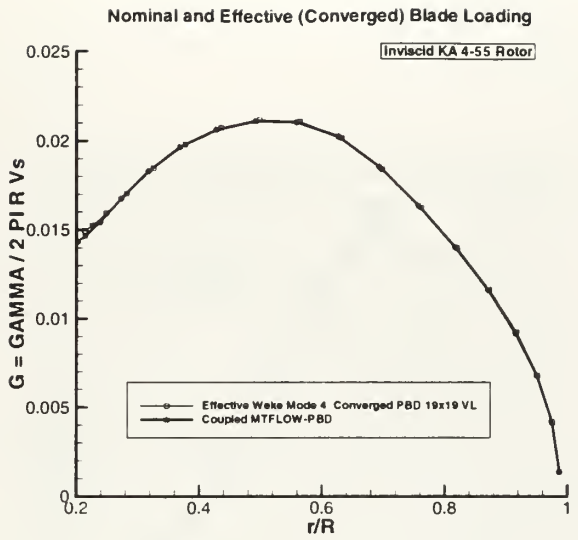


Figure 6-2: .
The PBD Mode 4 calculation was done with a 19x19 vortex lattice. This is the converged grid size. Since there is no vorticity (*i.e.*, no boundary layer) redistribution of the nominal wake, the predicted radial loading distribution from the effective wake calculation (*i.e.*, PBD Mode 4 run) and the coupled MTFLOW-PBD are identical.

examining the difference between the circumferential mean tangential velocity profile at the trailing edge of the blade that was output from PBD and the circumferential mean tangential velocity profile at the trailing edge of the blade that was output from MTFLOW. The throughflow code can not alter the tangential circumferential mean velocity . To do so would require adding or subtracting power from the fluid. Therefore, the radial distribution of tangential velocity output by the propeller code must be the same as that within the throughflow solver. To ensure consistency, they should be identical. In practice, however, if they are identical, too much swirl is introduced into MTFLOW, and the blade overpowers the flow. Hence, the circumferential mean tangential velocity profile output by MTFLOW should be about 3-5 percent lower than the PBD input. A sample comparison velocity profile is shown in Figure 6-3.

6.2 No Propeller, Viscous Validation

The boundary layer routines themselves were validated using tests listed in Table 6.2. These are all essentially stand-alone tests and independent of the flow solver and the propeller code. As they represent more of a code debugging effort than a theoretical development, the resulting plots are not shown.

6.2.1 Huang Body of Revolution Nominal Wake

The Huang body of revolution is an axisymmetric, unappended subsea notional test vehicle tested extensively throughout the 1970's and 1980's [9], [8] [7], [6]. There were a total of three models built, which differed in the after body buttock lines. The primary use of these models was to investigate the boundary layer characteristics in the stern region of the model with the three different stern geometries, with and without a propeller running. The model geometry for the so-called Huang Body 1 is shown in Figure 6-5.

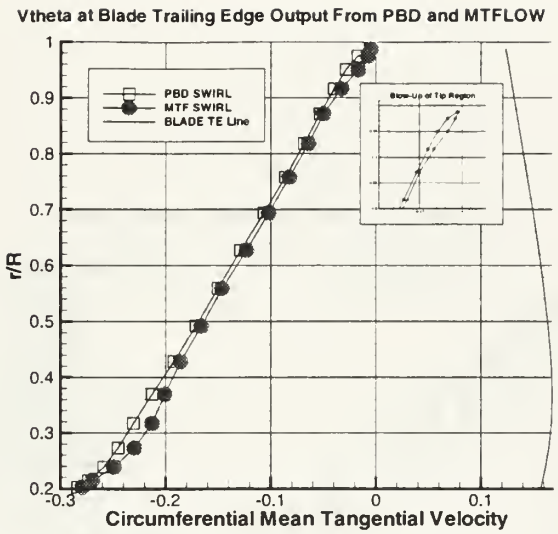


Figure 6-3: The circumferential mean tangential velocity at the blade trailing edge position output from PBD and MTFLOW. They are not equal because the blade circulation extends beyond the blade tip, overpowering the fluid. Hence the overall strength of the swirl (circulation) was decreased to account for this effect

.

1	Reconstruction of Swafford Profiles	The closure relations used in the MTFLOW formulation are provided by the Swafford profiles. The routines were required to exactly reproduce the Swafford profile for the test conditions indicated in [23].
2	Comparison to $\frac{1}{7}^{th}$ Power Law	Using the same inputs for the boundary layer, a profile was created using the $\frac{1}{7}^{th}$ Power Law and the Swafford routines.

Table 6.2: Boundary layer velocity profile validation tests.

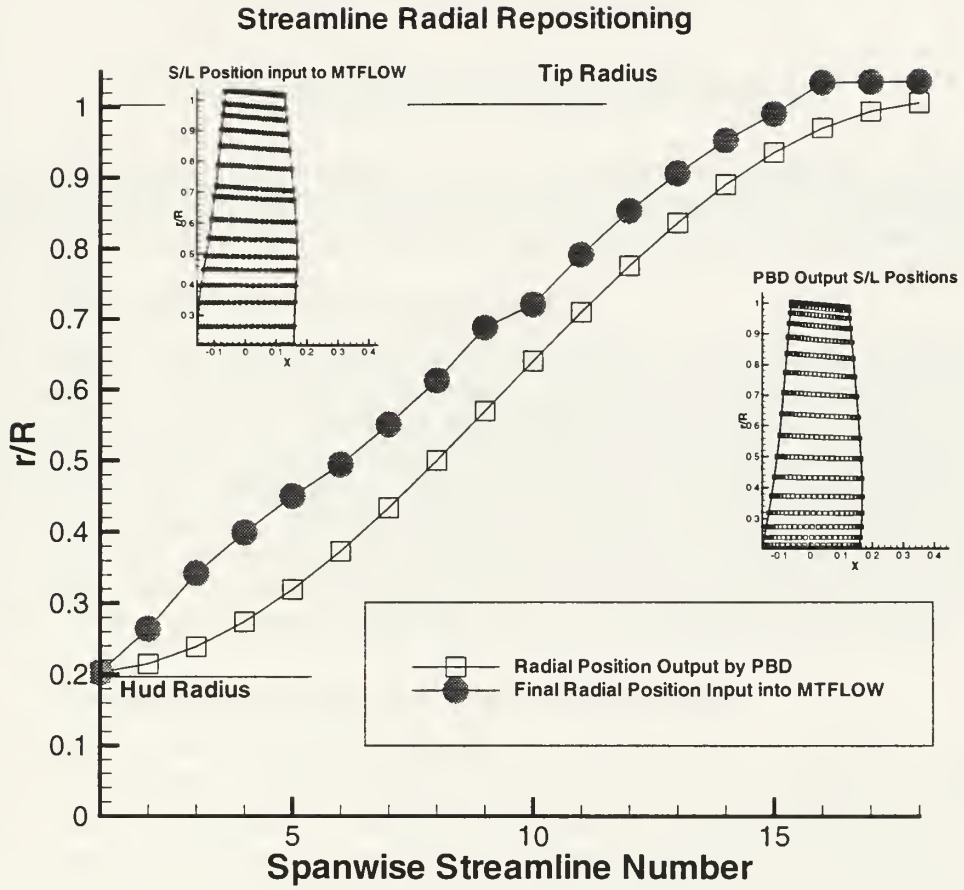


Figure 6-4: This plot highlights the non-uniform repositioning of the streamlines required to perform the nominal wake validation test successfully.

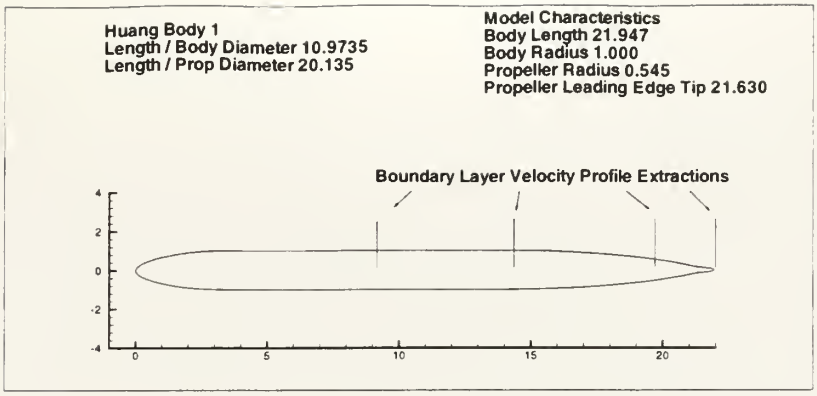


Figure 6-5: Huang Submarine Body with Stern 1.

Measurements carried out on Huang Body 1 included measurement of pressure along various stations, velocity profiles at multiple stations along the body with and without a propeller, and detailed measurements of boundary layer quantities necessary to refine boundary layer closure relations.

6.2.2 MTFLOW as a Viscous Calculator

While this thesis is concerned with the interaction of a propeller and free stream vorticity, MTFLOW was first validated as a suitable tool for analyzing the flow around an external body with no propulsor. As such, numeric tests were conducted on the unappended Huang Body 1 model to predict the nominal inflow velocity in the stern region of the body. The velocity profiles calculated from the present method with the IBLT and the Swafford reconstruction algorithm were compared with the results from two different RANS codes [24] and the empirical data, as tested by Huang [9]. The velocity cut is taken very near the stern of the body as shown in Figure 6-5. The comparison is shown in Figure 6-6. The discrepancies between the three numeric codes and the empirical data suggest that further refinements are possible in the realm of boundary layer modelling.

6.2.3 $\frac{1}{7}^{th}$ Power Law Boundary Layer Comparison

An inquiry was made into the general utility of using the complex IBLT formulation by comparing the results of the IBLT with a simple $\frac{1}{7}^{th}$ power law calculation of momentum thickness and displacment thickness along the Huang Body 1. Figure 6-7 shows that the $\frac{1}{7}^{th}$ power law calculation is very poor within the stern region, and any velocity predictions based upon this result would suffer accordingly. Also, a resistance prediction of the body based upon an analysis of the downstream momentum thickness would lead to an inaccurate thrust requirement for the propulsor.

The difference in predictions on the aft part of the body show the influence of neglecting the pressure gradient due to curvature in the streamwise direction, and the crossflow pressure gradients arising due to transverse curvature. The resulting drag, computed from the downstream momentum thickness, is an order of magnitude different.

6.2.4 Boundary Layer Velocity Extraction Validation

To test the boundary layer velocity extraction routines, boundary layer velocity profiles were calculated using the Swafford routine and a simple $\frac{1}{7}^{th}$ power law prediction. In fact, White [25] reccommends use of the $\frac{1}{7}^{th}$ power law for general applications. In the $\frac{1}{7}^{th}$ power law there is a simple relationship between velocity and distance off the boundary.

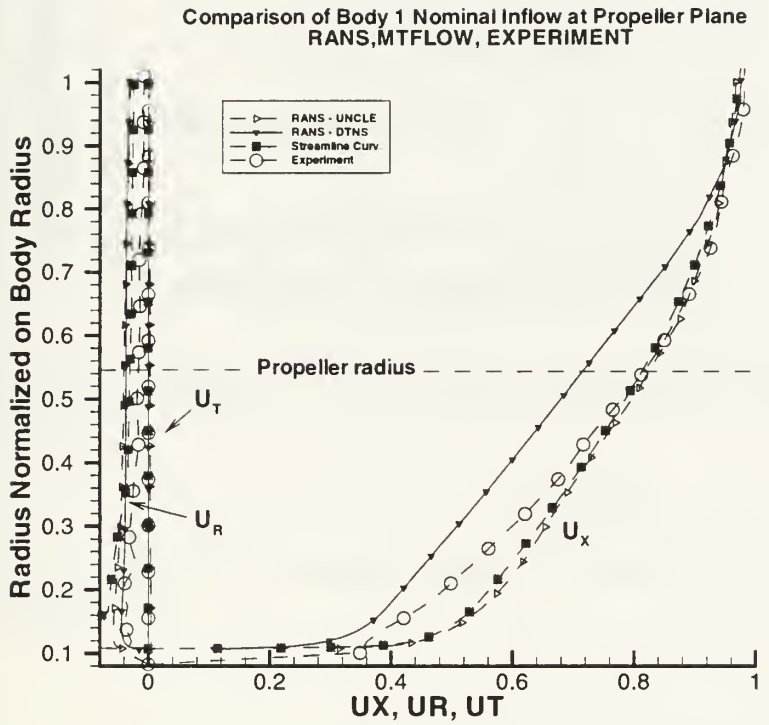


Figure 6-6: Comparison of MTFLOW, RANS, and Experiment Velocity Nominal Wake Profiles. Notice how well MTFLOW predicts the results in comparison to some of the RANS results, which take orders of magnitude longer to run.

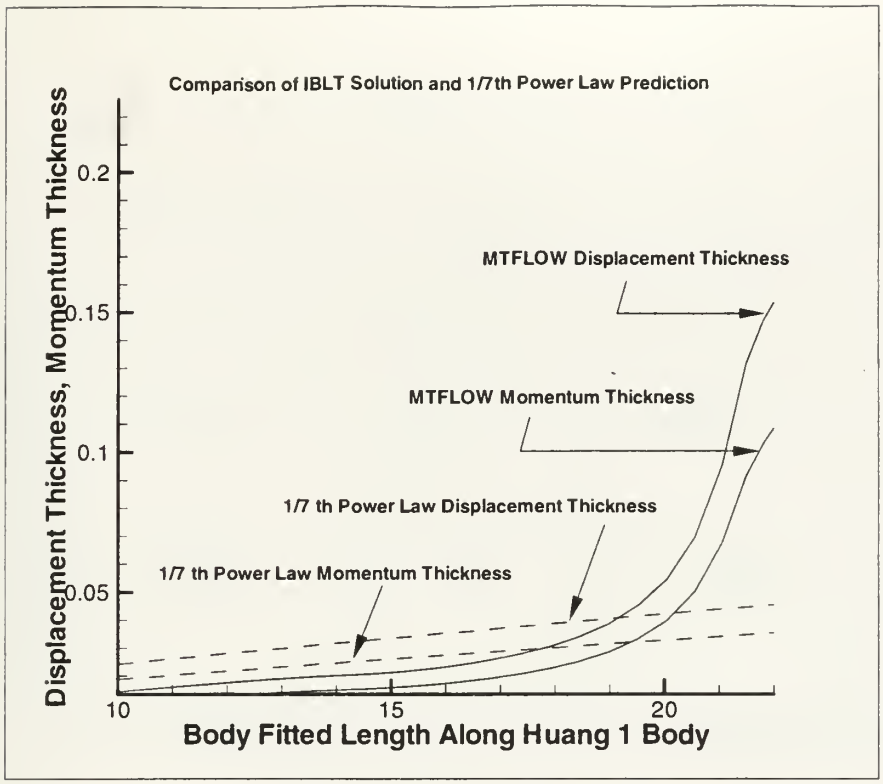


Figure 6-7: Comparison of MTFLOW IBLT predicitons and $\frac{1}{7}^{th}$ power law prediction for δ^* and Θ along Huang Body 1

$$\frac{u}{u_e} = (\frac{y}{\delta})^{\frac{1}{7}} \tag{6.1}$$

There is also a simple relationship between the boundary layer thickness and the displacement thickness [18].

$$\delta = 7.987\delta^* \tag{6.2}$$

Figure 6-8 shows representative velocity profiles predicted by the Swafford profile relations and the $\frac{1}{7}^{th}$ Power Law for the same input parameters of δ^* and u_{edge}. The body used is the Huang Body 1. The boundary layer quantities δ^* and u_{edge} were taken from the MTFLOW solution. Notice that far upstream of the propeller, where there are low pressure gradients, the $\frac{1}{7}^{th}$ Power Law prediction is in good agreement with the Swafford correlation. Near the propeller plane at the aft end of the Huang Body 1 model, however, there are large discrepancies, especially within the model propeller radius of 0.545.

6.3 Open Propeller, Viscous Validation

The best way to test the performance of the present coupling methodology is to validate against experimental data. Two test cases were chosen to validate the coupling procedure.

The first validation is the the Huang Body 1 tests with propeller 4577 operating at a constant J value. This represents an extremely hard case, because the propeller operates entirely within the boundary layer of the body. This requires a robust boundary layer method.

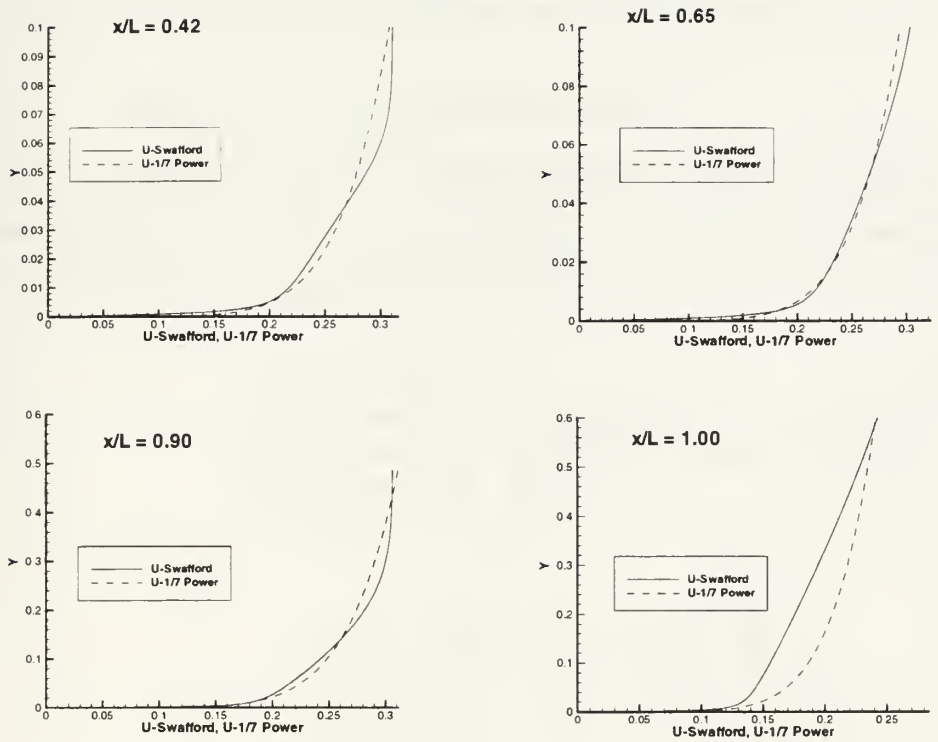


Figure 6-8: .

Comparison of Swafford Profile and $\frac{1}{7}^{th}$ Power Law Boundary Layer Velocity Profiles. The location of the stations are shown in Figure 6-5. The same boundary layer quantities predicted by the IBLT were input into a velocity reconstruction routine based upon the Swafford data, and one based upon the $\frac{1}{7}^{th}$ Power Law.

J=1.25	Experiment	Current Technique
K_T	0.2276	0.2164
$10K_Q$	unk	0.4685

Table 6.3: Huang Body 1 with Propeller 4557: experimental and numerical results.

The second test case is the 1998 ITTC Propeller Workshop conference test case of the three bladed propeller 4119 on a downstream driven shaft. This case was chosen to allow comparison with the validation performed by ITTC. This canned test case really tests the interaction of the propeller and the inviscid flow, since the boundary layer is confined to a very thin region close to the shaft, and presents an opportunity to see how the coupling procedure performs under real world conditions.

6.3.1 Huang Body 1 with Propeller 4577

The stern section of Huang Body 1 with the projection of propeller 4577 is shown in figure 6-9. The gridlines in the Figure are gridlines, not flow streamlines, or finite volume cells. Figure 6-9 highlights the complexity of the propeller in the computational domain due to the non-cylindrical propeller sections, non-radial propeller generator line, and non radially uniform inflow field.

The particulars of propeller 4577 are shown below.

Number of Blades	7
Section Meanline	NACA a=0.8
Section Thickness	DTNSRDC modified NACA 66
Rake Angle (deg)	6.964
Skew (deg)	30

$\frac{r}{R_P}$	$\frac{Chord}{D_P}$	$\frac{t}{C}$	$\frac{P}{D_P}$	$\frac{f_M}{C}$
0.2106	0.171	0.235	0.823	0.0014
0.3000	0.177	0.209	0.980	0.0175
0.4000	0.182	0.182	1.151	0.0288
0.5000	0.185	0.158	1.243	0.0337
0.6000	0.185	0.135	1.264	0.0341
0.7000	0.180	0.116	1.248	0.0311
0.8000	0.164	0.0995	1.206	0.0246
0.9000	0.132	0.0875	1.157	0.0141
1.0000	0.069	0.0813	1.108	0.0000

The results from the coupling closely match experimental results, as shown in Table 6.3.

6.4 Propeller 4119: ITTC 1998 Tests

In Spring 1998, the International Towing Tank Committee (ITTC) held a propeller workshop for RANS codes and panel methods. The premise for the workshop was to validate various numerical codes against empirical test data for propeller DTMB 4119 tested in the Naval Surface Warfare Center, Carderock Division 24 inch water tunnel [10].

The test parameters for a single advance coefficient are shown below.

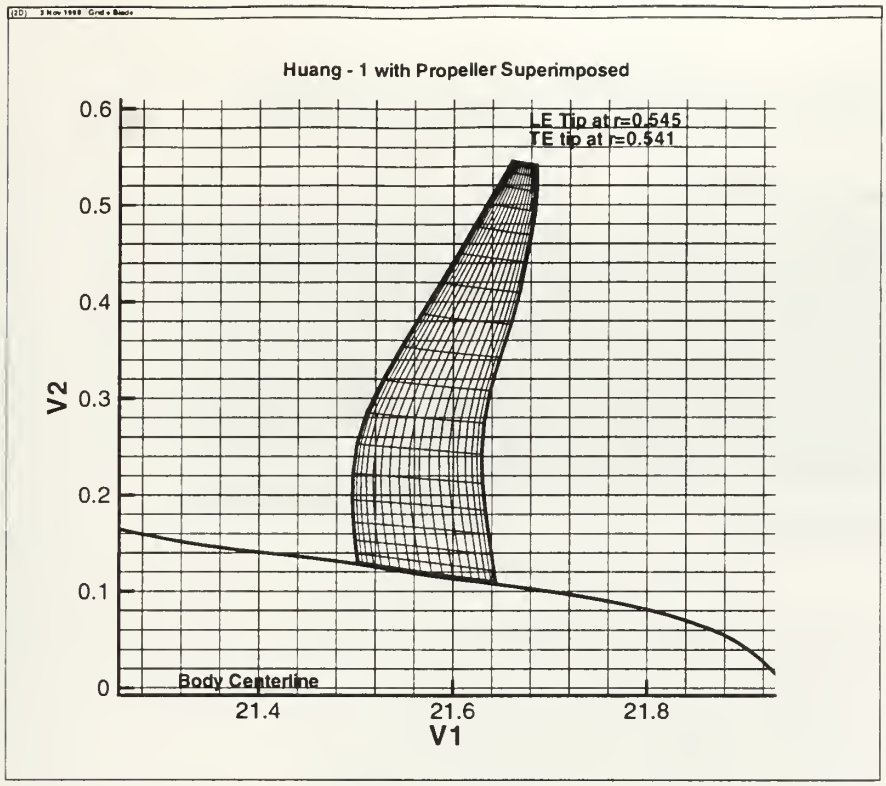


Figure 6-9: Huang Body 1 Stern Profile with Propeller 4577

$J = \frac{V}{nD}$	0.83333
V, advance velocity	8.33 $\frac{ft}{sec}$
n, rotation speed	10 rps
D	12 in
R	6 in
$\frac{V_{inlet}}{V}$	1.0093

Note that the inflow speed at the test section inlet is not 1.0 . This required calculating an effective advance coefficient as described in Section 3.3.3.

The following flow parameters were used within MTFLOW.

Inlet Mach Number	0.001488
Reynold's Number	432000
Far Field Type	3

The MTFLOW grid is shown in Figure 6-10. Grid is actually a misnomer, since the gridlines are actually the streamlines. As such, the initial grid represents the initial guess at the streamline positions. Streamlines were bunched near the shaft only to resolve the potential interference between the leading edge separation bubble and the propeller blade. The truncated grid domain was used to overcome MTFLOW spline interpolation problems.

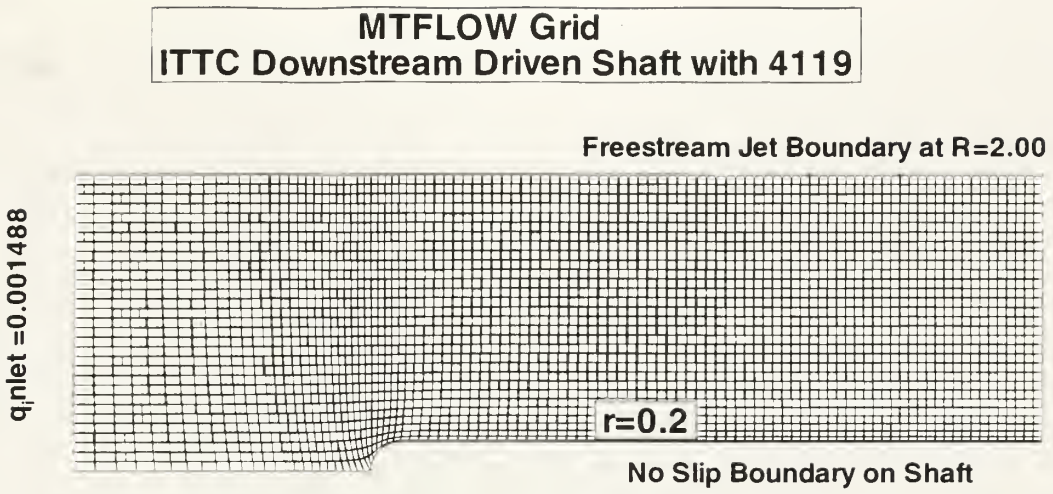


Figure 6-10: Propeller 4119 in ITTC Configuration

J	Exp K_T	Present K_T	Exp $10K_Q$	Present $10K_Q$
0.700	0.200	0.194	0.360	0.347
0.833	0.146	0.141	0.280	0.269
0.900	0.120	0.118	0.239	0.237
1.100	0.034	0.045	0.106	0.131

Table 6.4: Experimental and numerical propeller curve test results for 4119.

6.4.1 Inviscid Comparison to Experimental Data

An inviscid coupling was performed to validate the overall coupling methodology and investigate the magnitude of the radial streamline redistribution problem.

The published experimental results for the Workshop test case and the numerical results from the inviscid PBD-MTFLOW coupling are shown in Table 6.4.

For this series of test, the linear redistribution of radii was used, vice the non-linear redistribution used in the KA455 test. These results show that the simplfied, linear redistribution is satisfactory to reproduce experimental results at lower J values. At high J the linear model results in too much loading at the tip, which results in too high K_T and too high K_Q. Of note, the calculation only took 2 hours to converge all data points.

6.4.2 Problems and Solutions

The biggest problem was that MTFLOW had a lot of problems creating a two dimensional spline representation of the input swirl. A sample spline interpolation problem is shown in Figure 6-12. The final solution was to alter the gridding until a "happy medium" was found between the PBD output swirl distribution and the MTFLOW grid. There did not appear to be a rational, consistent method to resolve the splining problems.

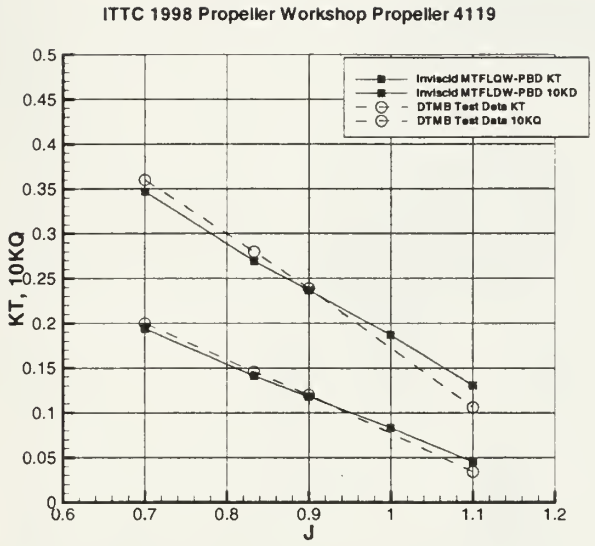


Figure 6-11: comparison of MTFLOW-PBD with P4119 ITTC Test.

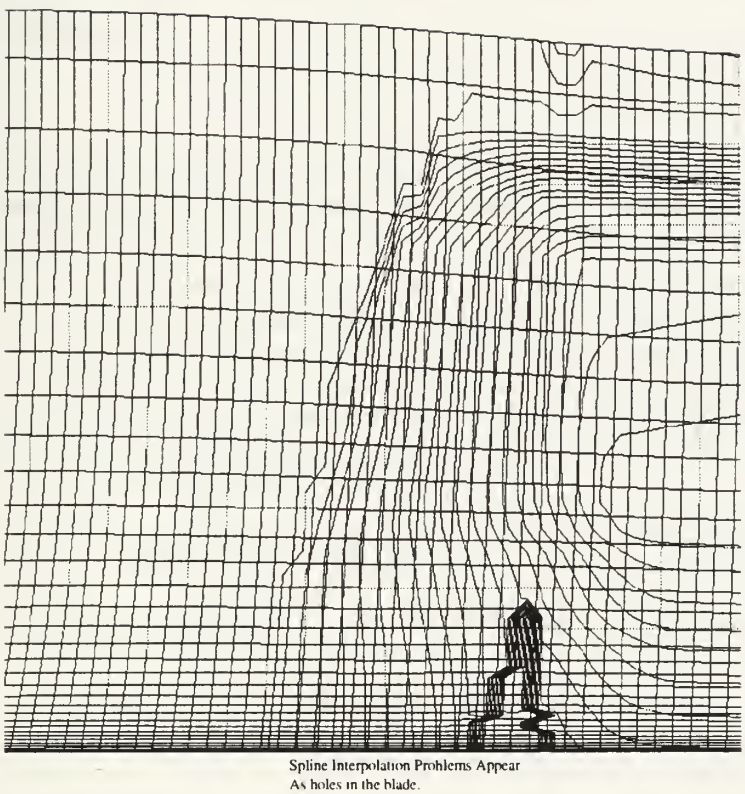


Figure 6-12: This is a view within MTFLOW of a contour of the swirl input from PBD into MTFLOW. Notice the large jagged peak on the hub radii. This is the symptom of the spline interpolation problem. When the MTFLOW spline routine fails, it puts a 0 data point in the field. Thus, the flow field responds as if there were zero swirl in the flow at that point – a porous propeller.

Chapter 7

MTFLOW-PBD 14 Fan Design Mode

To test the robustness of the coupling methodology, the PBD-MTFLOW coupling procedure was applied to the analysis, re-design, and subsequent analysis, of a ducted fan unit. The design of a ducted fan presents several technical challenges:

1. The high contraction ratio from the inlet to the fan section and vertical walls present in the transition section present difficulties to the flow solver. Actually, within PBD 14.4 it is impossible to extend the hub and duct vortex lattice system across a vertical boundary since all velocity data is interpolated via a cubic spline with respect to the axial location. On a vertical boundary, there are are multiple flow values at the same axial location. This situation is termed a multi-value solution and can not be resolved via the interpolation methodology as currently implemented within PBD 14.4.

2. The very small radius of curvature duct lip just upstream of the leading edge makes the enforcement of the kinematic boundary condition within PBD on the duct difficult.

3. Low inlet speeds can result in negative effective inflow velocities. Thus, the fan blades are fighting to overcome their own negative effective inflow.

The particular fan unit design was motivated by a design competition sponsored by a local fan design firm [21]. The sponsor set the design rotation speed, flow rate, hub ratio, and duct geometry. All length scales were scaled such that the leading edge tip radius of the fan was 1.0 to be consistent with PBD scaling.

7.1 Fan Radiator System Geometry

The fan was designed to operate as part of a radiator unit, with the radiator upstream of the fan. The geometry is shown in Figure 7-1. This geometry was modelled in MTFLOW with the exception

Rotation Speed	2300 rpm
Mass Flow Rate	0.419 $\frac{kg}{sec}$
Hub Radius	0.40
Tip Radius	1.00
Length Scale	1.46

Table 7.1: Fan Design Operating Parameters.

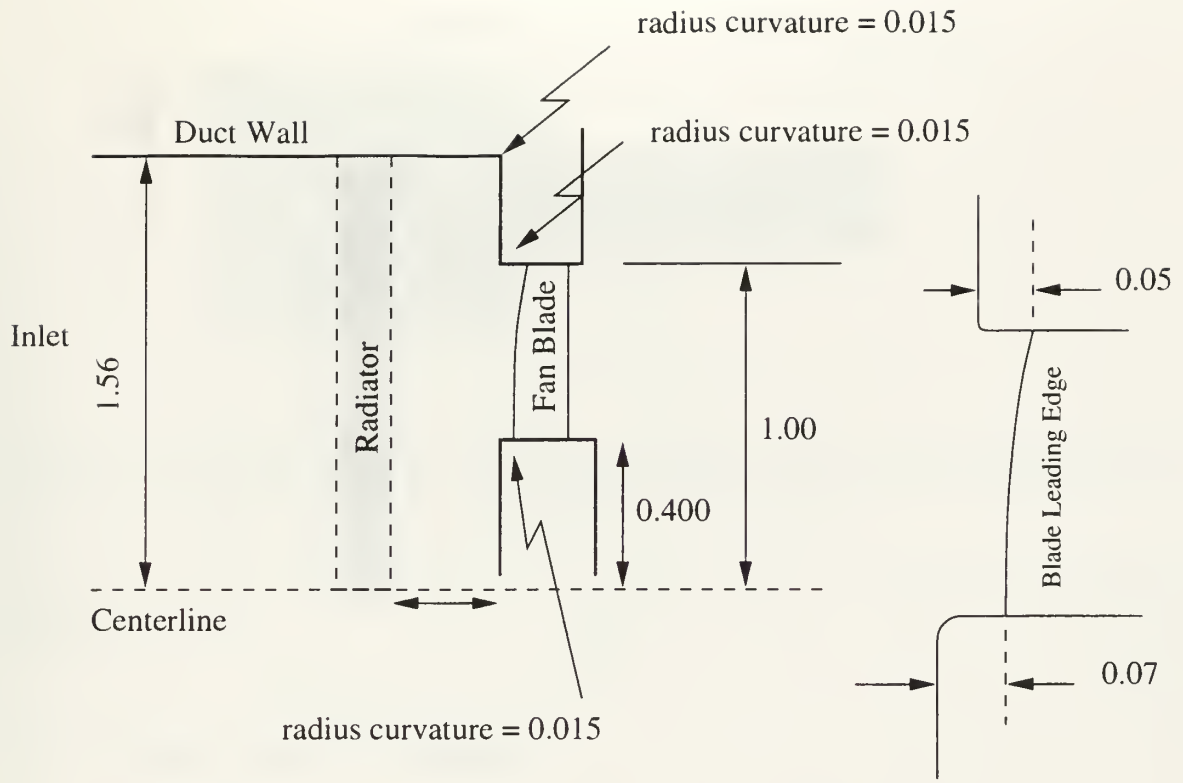


Figure 7-1: The geometry of the radiator fan system. This geometry was modelled in MTFLOW with the exception that the downstream hub and duct lip was modelled as a straight horizontal wall. Also, the radiator was not modelled.

that the downstream hub and duct lip was modelled as a straight horizontal wall extending to infinity. This was done only to conserve computational expense. Physically, the high flow region around those lips is located too far downstream of the fan to interact with the fan blades. Also the radiator was not modelled. The effect of the radiator is to dump enthalpy into the fluid, and straighten the flow. So, the assumption is that the radiator is at infinity, and produces no change in enthalpy.

The engine designer desires the shortest length fan possible to reduce engine "stack" length. In fan terms, we want the lowest possible pitch angles, such that fan stack length is not reduced by reducing the chord lengths of the fan blades. A reduction in chord lengths is undesirable for improved off-design performance.

The final MTFLOW grid used is shown in Figure 7-2. The long downstream run is required by PBD to grow the trailing wake system. Final solution results showed little change in the flow characteristics downstream of one propeller radius. The effects of modelling a shorter wake within PBD were not investigated. The upstream length of the domain is set by the requirement that the inviscid streamlines correctly match the constant inflow assumption.

7.2 Initial Blade Design

The initial blade design was carried out by Dr. Robert VanHouten using his own two-dimensional blade section design code. Due to the internal geometry limitations of the code used, the original

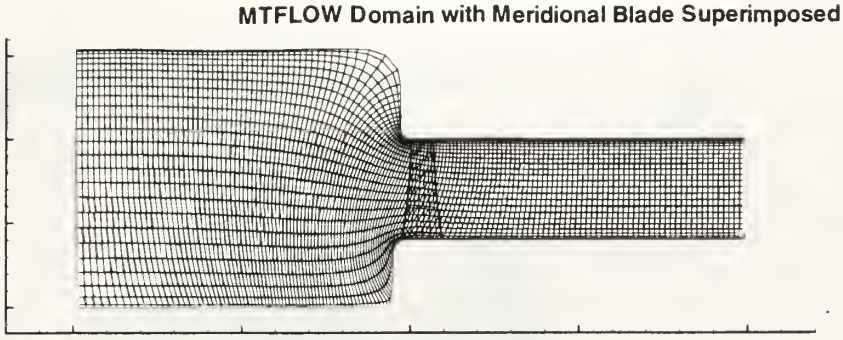


Figure 7-2: The flow grid used within MTFLOW. The long downstream extent is required by PBD to grow the trailing wake system. The upstream length of the domain is set by the requirement that the inviscid streamlines correctly match the constant inflow assumption.

fan design consists of seven radial blades. The initial fan design by Dr. Van Houten is shown in Figure 7-3.

7.3 Radial Blade Analysis

The initial radial blade was analyzed using PBD-14.4 with a vortex-lattice hub and duct representation, coupled with MTFLOW. The blades were discretized using a vortex lattice mesh of 225 elements. Similarly, the hub and duct were discretized upstream and downstream of the propeller. The hub and duct vortex lattice systems were not extended over the entire vertical sections upstream of the fan due to internal numerical limitations noted in section 7.

The result of this inviscid analysis was a solved-for radial circulation distribution along the span of the blade. A comparison between the radial circulation distribution predicted by PBD14.4-MTFLOW and the original design circulation distribution along the span of the blades is shown in Figure 7-4.

The final converged flow solution streamline mesh and axial velocity contours are shown in Figure 7-5. The extremely high inviscid meridional velocity at the duct lip near the tip of the blade can lead to instabilities in converging the flow solution. This high velocity highlights one of the limitations of the current coupling technique. While the throughflow is for the circumeferential mean flow, the blade experiences a different local blade velocity, due to the three dimensional nature of the blade vortex sheets. While it is well-known that the three dimensional unsteady harmonics of the propeller solution attenuate rapidly away from the propeller, in the flow problem considered here, it is not obvious that there would be no interaction between the local, three dimensional, unsteady harmonic solution and the meridional flow which, due to the high velocity gradients in this region, may produce undesired loadings or blade fluctuations.

7.4 40^o Back Skew Blade Design

The first design iteration was to design a fan blade with the same circulation distribution as the radial fan blade, but with the addition of 40^o of backskew to the blade tip. The back skew serves two purposes:

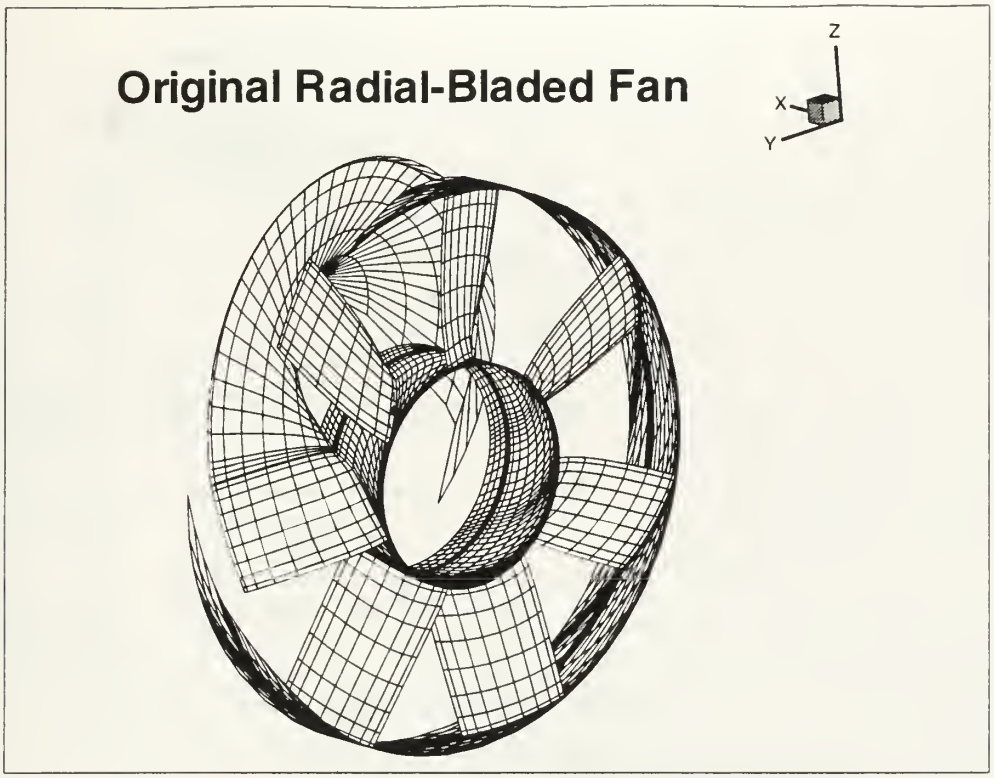


Figure 7-3: The original radial fan with cut-away duct. The trailing wake vortex system is shown emanating only from one blade to maintain clarity.

Reduce non-circumferential mean effects . Quite often the propeller blade operates in a flow domain that is non-uniform. Usually this is due to an upstream obstruction or contorted flow passage. As the blade passes through this region of fluid with a different velocity, the apparent angle of attack of the blade is changed. This is shown in Figure 7-6. The effect of this change in angle of attack is to change the lift produced by the blade, which results in a non-uniform pressure wave - noise! . If the change in the angle of atack is large enough, the blade can stall and vibrate horribly. If the entire blade leading edge passes through this region of non-uniformity simultaneously, large unsteady pressure fluctuations result. In english, this means that a lot of noise is generated. If, however, the blade is skewed such that the blade sections individually experience the change in angle of attack, the strength of the unsteady pressure wave is reduced, and much less noise is generated.

Reduced the pitch angle of the blade tip . The effect of the skew is to reduce the pitch angle of the tip section. This is important in fan applications where the meridional blade profile is a premium, and any reduction in the length of the meridional blade sections positively impacts the overall system design.

The design process proceeded as follows:

1. An initial-guess blade shape is designed with the desired skew distribution. In this case the distribution specified a tip skew of 40 degrees.

2. By holding the leading edge line fixed, the propeller blade design code alters the position of the rest of the blade points to find the blade shape required to satisfy the kinematic boundary condition and produce the required loading.

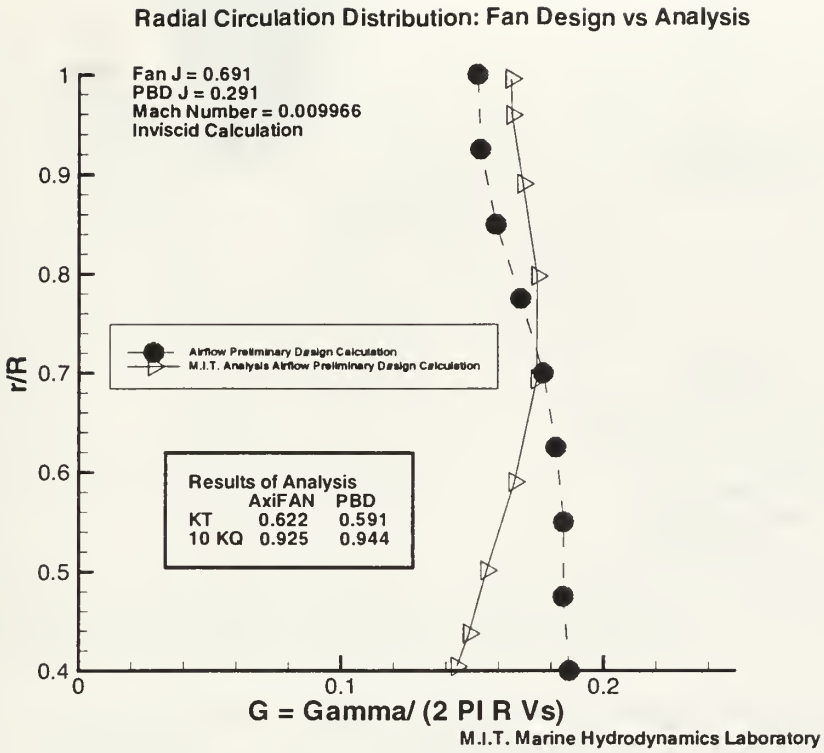


Figure 7-4: The design radial circulation distribution and final PBD14.4-MTFLOW analysis of the same blade. The blade was designed using PBD14.2 which uses an image representation of the hub and duct. Note the discrepancies in the tip and hub region. This is most likely due to the PBD14.4 vortex lattice representation of the hub and duct.

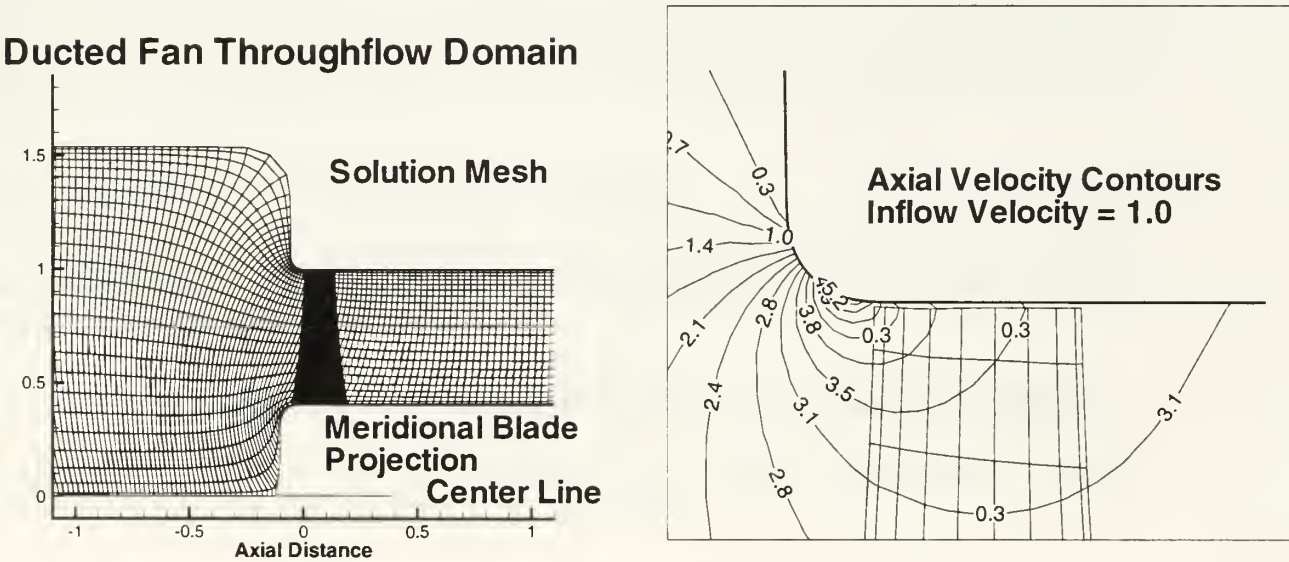


Figure 7-5: These two plots show the final MTFLOW inviscid solution streamlines. The plot on the right is a blow-up of the propeller tip region. Notice that the propeller leading edge tip operates in the extremely accelerated flow round the duct lip. This is a purely inviscid flow effect.

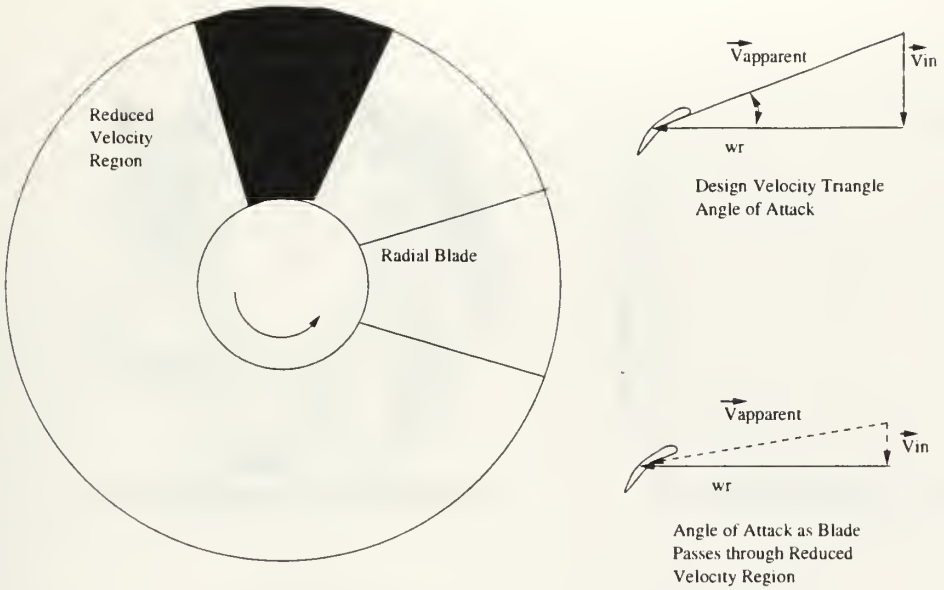


Figure 7-6: As the blade passes through a region of inflow with reduced velocity, the velocity triangle clearly shows the accompanying increase in angle of attack.

3. Because the meridional distribution of swirl has changed, this new swirl distribution is input into MTFLOW, and a new circumferentially mean (i.e., meridional) velocity distribution is found, which satisfies the fluid equations of motion.

4. The updated velocity field is extracted from MTFLOW and input into the propeller blade design code, where a new blade shape is found. That is, the process loops back to Step 2.

This process continues until the root mean square (rms) of the change in blade shape is less than a user-specified tolerance.

In plain propeller designer English, the main effect of the blade design code is to find a suitable pitch distribution (*i.e.*, set a local sectional angle of attack on the blade section), given the desired skew distribution. The camber distribution across the chord is specified by the user and can not be altered.

7.5 40^o Forward Skew Blade Design

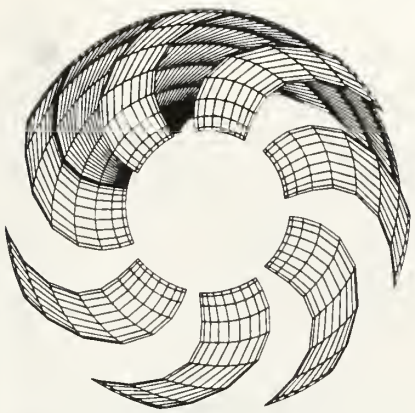
A forward skewed blade was designed in the same manner as the backward skewed blade. Starting from the desired circulation distribution, the blade was skewed such that the tip was skewed to a forward, or negative, angle of 40 degrees.

7.6 Comparison Between the No Skew and Skewed Blades

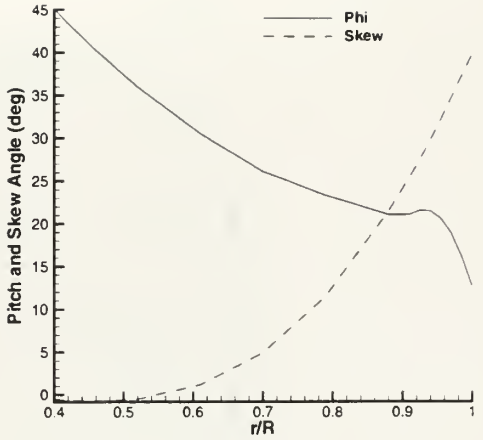
The plots shown in Figure 7-7 show the effects of skew upon the resultant blade shapes. The most noticable change is in the pitch angle of the blade tip region.

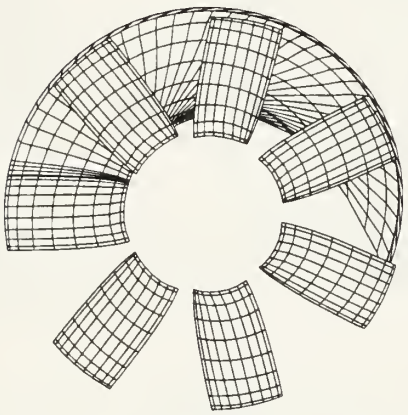
Figure 7-8 dramatically highlights the pitch and camber redistributions introduced by the skewing of the blade.

The only section properties not investigated were the chord and raked distribution. Unlike a marine propeller which needs long chord lengths to prevent cavitation, an airscrew uses longer chord

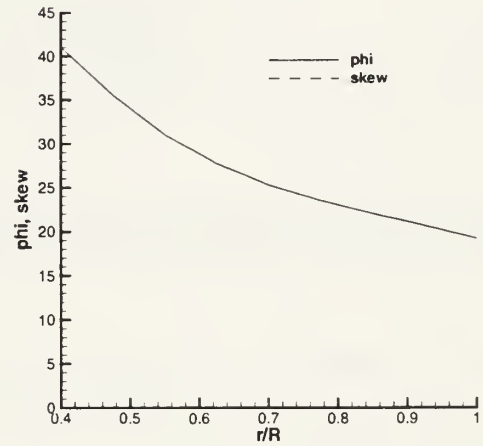


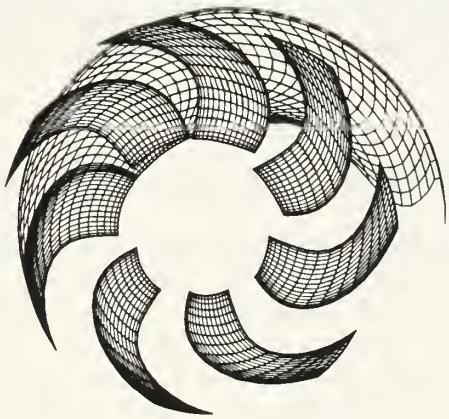
40 Degree Back Skew





0 Degree Skew





40 Degree Forward Skew

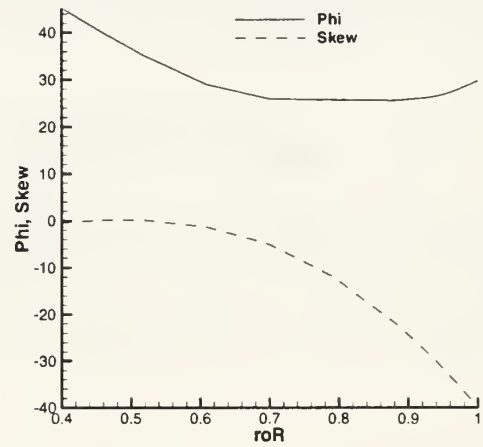


Figure 7-7: The final fan designs for the back skew, no skew, and forward skew fans. The small plot to the right of the fan shows the distribution of pitch(phi) and skew angle over the radius of the blade.

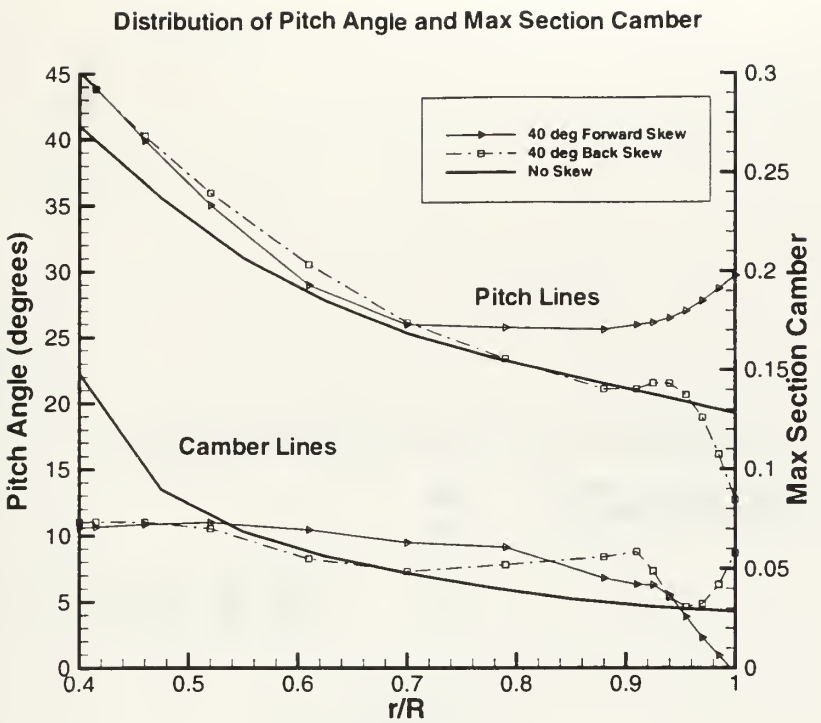


Figure 7-8: The rake and max section camber for the final fan designs. Notice how the changes in section pitch (*i.e.*, changes in local angle of attack) are accompanied by an offsetting change in section max camber.

lengths to improve unsteady performance. The effects upon performance, specifically with regards to boundary layer separation, were not investigated in this thesis.

Finally, rake is used by some designers as a complement to skew to introduce a non-uniform leading edge span line to any inflow unsteadiness. In the ducted fan case considered above, the application of rake to the de-pitched back skewed blade would enhance the fan performance, since the leading edge tip would be moved further away from the high velocity region of the duct lip.

7.7 A Re-Analysis of the Skewed Blade Design

A re-analysis of the skewed blade designs revealed that there may be problems with the PBD fan design mode. Specifically, the use of images to represent the duct in the design, led to an inaccurate flow representation in the tip region. With the highly skewed blades, the old paradigm of extending the image vortex lattice system radially outward from the tip breaks down, because a significant portion of the fan blade does not lie along a radial line exending inward from the tip. The end result is that the image vortex lattice system produces too weak an effect upon the rest of the blade, and the blade is overloaded.

This problem is highlighted after a re-analysis of the designed for blade was done with a vortex lattice representation of the hub and duct. With a vortex lattice duct, the kinematic boundary condition is exactly enforced to produce the blade to blade variational flow. From Figure 7-9 we see that the blade designed using the image method is overloaded and will probably stall.

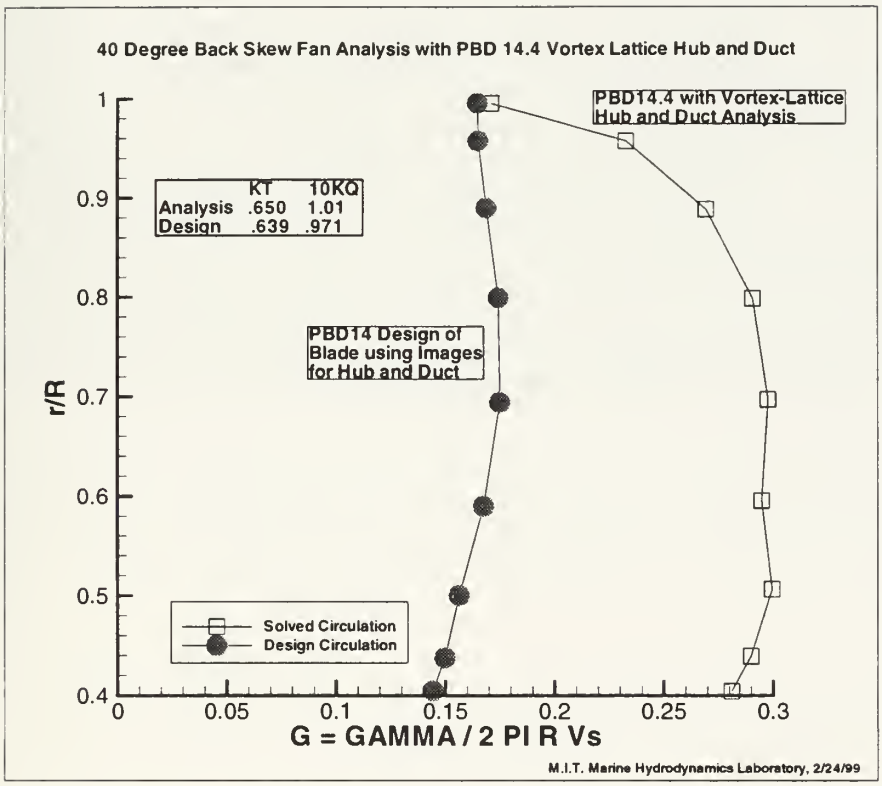


Figure 7-9: The curve on the left represents the spanwise loading distribution predicted by PBD using an image hub and duct representation. The curve on the right represents the spanwise loading distribution predicted by PBD using a vortex lattice hub and duct representation.

7.7.1 The Back Skew on the Test Stand

The 40 degree back skew fan was built and tested by Airflow Research. At the design speed, the fan stalled due to the extremely aggresive pitch of the blades. When the fan was powered down, a tremendous amount of airflow was produced, with an attendantly high torque. Although no data was actually collected, observation confirmed that the vortex lattice hub and duct predicted results were more in line with reality.

7.8 Specific Problems of the PBD Fan Design Mode

The largest noted problem is the fact that PBD internally splines all flow quantities with respect to the axial location. In highly contracting flow domains, such as this ducted fan case, where there are nearly vertical boundaries, splining with respect to x leads to a multi-valued solution. In english, the resulting 3^{rd} order polynomial oscillates wildly. Any interpolations taken from such a polynomial will in no way represent the actual fluid properties at the interogated point in space.

One solution to this problem is to spline with respect to the distance along a streamline. The difficulty then is how to interpolate to find the flow velocities at an arbitray point. For instance, for an arbitrary point, it is impossible to *a priori* know which streamline the point lies along, and the streamwise distance along the streamline at which the point lies.

Another proposed solution is to transform the flow domain from an $x-y$ coordinate system to a general curvi-linear $\xi-\eta$ coordinate system. Then a spline with respect to ξ will not be multi-valued.

A second noted problem is in the pitch angle and number of vortex lattice elements placed upstream of the blade leading edge tip. When a the tip of the propeller has a low pitch angle, as in the case of the 40° back skew fan, the current implementation of making the vortex lattice follow the pitch line of the blade leading edge results in a highly skewed lattice.

Chapter 8

Conclusions

MTFLOW coupled with PBD works. This thesis has shown, through the nominal and effective wake calculation comparison, that the coupling is based upon sound principles, and that there are no fundamental reasons that PBD and MTFLOW should not be coupled. As such, MTFLOW represents a viable alternative to axisymmetric RANS codes for solving circumferentially mean throughflow problems, because of the large reduction in computational cost.

The devil, though, is in the details. More refinement is needed in handling the streamline radial redistribution problem. But now that the coupling codes exist, these are refinements upon an established base.

8.1 Propeller Blade Design Code Improvements

The steady design and analysis propeller code, PBD, should be continually updated to reflect the increasing sophistication of preliminary designs. Because the blade singularity discretization leads to a requirement to couple PBD with some sort of throughflow solver, time should be spent on creating common data models which can easily be manipulated by PBD and any generic throughflow solver such that information is not lost in the translation.

8.1.1 Analysis Mode

The analysis capabilities of PBD are extremely robust. The extended vortex lattice hub and duct system, however, increases the computational time associated with the solution. This increase in time becomes noticable when coupling PBD with an extremely fast throughflow solver such as MTFLOW. More attention to the underlying numerical data representation, and numerical solution techniques, with an eye towards speed, would propel PBD back ahead of the pack.

There may be issues with the current implementation of the hub and duct vortex lattice model. One issue is the increase in computational time required due to the larger matrix system involved. Another issue is the lack of user control over the orientation and placement of the lattice grid. Highly stretch grid lattices may result from various geometries. Perhaps a solution to these problems may be a B spline hub and duct representation.

8.1.2 Design Mode

Attention should be focused on incorporating a vortex lattice hub and duct in the design mode to account for the effects of blade skew and radially non-uniform hub and ducts. Perhaps the only solution will become, then, design by analysis. Perhaps, though, there is the elegant solution lurking just around the corner.

8.2 Boundary Layer Modelling Improvements

This thesis highlighted the disimilar boundary layer profiles which result from the use of present state of the art RANS and IBLT boundary layer models. The fact is that the question is still open on if a generally applicable, turbulent boundary layer model exists. An added dimension of complexity is the proper treatment of the propeller induced body forces in the turbulent boundary layer closure equations. Such research may lead into a better understanding of flow around submerged bodies, and better predict the propeller inflow velocity field.

Appendix A

User Notes on Running MTFLOW

A.1 Problem Solving

When problems are encountered with MTFLOW, it is best to look at the contour plot within MTSOL of the state parameters ρ, q, and P. It is also useful to look the MTSOL contour plot of any variables that you input to check for spline problems. Spline problems manifest themselves as large or small holes in the solution. I found that the velocity contout plot (MTFLOW q variable) was helpful to identify trouble areas.

A.2 Grid Issues

The interpolation of the input swirl into MTFLOW is somewhat dependent upon the grid. While arbitrarily increasing grid density will not usually solve interpolation problems, clustering t lines near the problem area can sometimes fix the problem.

A.3 Input Swirl: $r\Delta V_\theta$

A.3.1 Interpolation

It is critical to adequately refine the swirl input near the blade tip. This is increasingly important when several radii are redistributed by PBD2MT near the tip. This require bunching several lines of constant t near the tip radius.

A.3.2 Convection

The user must manually convect any excess swirl downstream. If this downstream convected swirl encounters a downstream mechanical device that adds or removes swirl, the operator must combine the upstream and local swirl in a logical manner. The solution that I used was to first build the tflow input file for the downstream blade row in the usual manner and then convect the swirl from the trailing edge of the upstream blade row along the previous iteration streamlines through the downstream blade row. Remember that as the swirl convects down streamlines that change radial distance off the centerline of the machine, the $r\Delta V_\theta$ must be changed accordingly, since internal to MTFLOW the V_{theta} is calculated by dividing this quantity by the local radius.

A.4 Open Propellers

A.4.1 Far Field Setting

Assume a constant pressure jet boundary and set the Far Field singularity equal to 0. Inside MTSOL, set $FF = 3$. While this may cause a slight degradation of the flow boundary condition, my experience has shown that using a singularity causes more problems, especially when modelling a domain with an upstream or downstream shaft.

A.5 The Mach Number in Water

When running MTFLOW, the user specifies a flow Mach number and Reynold's number. While the Reynold's number is widely used in marine fluid dynamics, the use of mach number, in a nominally incompressible fluid, warrants a moment of thought. For a general case of interest, start from the known Reynold's number of the problem.

$$Re = \frac{VL}{\nu} \tag{A.1}$$

Knowing the length scale L and assuming a kinematic viscosity, the velocity scale is determined.

$$V = \frac{Re\nu}{L} \tag{A.2}$$

If it is now assumed that the speed of sound in water is nominally 1680 $\frac{m}{s}$ [25], the mach number for the flow problem is known.

$$M = \frac{Re\nu}{1680L} \tag{A.3}$$

Since the speed of sound in MTFLOW, designated as a in MTFLOW, is always 1.0, the mach number calculated from equation (A.3) is directly input into MTFLOW as the inlet mach number.

Appendix B

Modifications to MTFLOW Source Code

Modifications to the MTSET/MTSOL/MTFLO family of programs was limited to the mtset.f and io.f subroutines.

B.1 Modifications to mtset.f

Within mtset.f, an output routine was added to output the original gridlines calculated by MTSET. This information is required when BL2BODY lays the calculated boundary layer velocity profile back into the flow domain.

B.2 Modifications to io.f

The io.f subroutine was modified to output the integral boundary layer quantities and the flow solution with grid streamline positions. It has been validated for both external propellers and internal flow cases.

The following output files are written.

SUBROUTINE	ALTERATIONS
mtset.f	output ORIGGRID.tec, original grid set by MTSET, over the solid wall body points (downstream of the leading edge)
io.f	output OUTGRID.tec, final grid from MTSOL solution output OUTVEL.tec, cell-centered velocities from MTSOL output OUTBL.tec, boundary layer quantities in boundary layer coordinates

Table B.1: Custom MTFLOW output file in tecplot formatted ascii for tecplot viewing and use by BL2BODY

SUBROUTINE	ALTERATIONS
io.f	output OUTGRID.tec, final grid from MTSOL solution. It is useful for ensuring that BL2BODY routines did a good job in accurately reconstructing to flow domain. output OUTVEL.tec, This file has the z, r location for the $u, v, \Delta r V_\theta$ velocity components. It is cell- centered velocity data, like most RANS finite volume solver output data. output OUTBL.tec, boundary layer quantities in boundary layer coordinates

Table B.2: Custom MTFLOW output file in tecplot formatted ascii for tecplot viewing and use by BL2BODY

Appendix C

ITTC Propeller 4119 Input Files

C.1 PBD Input File

```
pbd-14.4 analysis of p4119 - coupled MTFLOW analysis
P4119.BSN                                : FILE NAME FOR BLADE B-SPLINE NET
restart.vel                              : FILE NAME FOR WAKE FIELD
 3 15 15                                 : nblade nkey, mkey
 1                                       : ispn (0=uniform,1=cos)
 14 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 :mctrp,mc(n)
 7 0.0 0 0.0                             : ihub,hgap,iduc,dgap
 8 6                                     : HDWAK, NTWAKE
 0.84 300                                : Cq, MXITER
 0.3 0.3                                 : OVHANG(1), OVHANG(3)
 10 -10 0 0                              : nx,ngcoeff,mltype,mthick
 6                                       : imode
 0                                       : nwimax
 1 0.001 0.00 0.0 1                      : niter,tweak,bulge,radwgt,nufix
 1 0.02                                  : nplot,hubshk
 4 2                                     : NOPT, NBLK
 1.11023  1.000   1.500   0.02           : ADVCO XULT XFINAL DTPROP
 0.00000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.000    : G
 0.20000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000 0.9000 0.9500 1.0000   : r/R
 0.06576 0.0563 0.0477 0.0396 0.0321 0.0250 0.0183 0.0120 0.0089 0.00000  :T/D
 0.00814 0.0086 0.0085 0.0084 0.0082 0.0081 0.0080 0.0079 0.0078 0.0079   : CD
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000              : ua
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000              : uau
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000              : ut
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000              : utu
```

C.2 MTFLOW Input Files

C.2.1 Coupling Admin File

```
432000                          ! Reynolds number
0.001488                        ! inlet mach number
1.0093                          ! Vship used in PBD to find J
15                              ! aft most point of body
4.000                           ! x location of LE tip
1.000                           ! r location of LE tip
tflow.4119                      ! tflow file name
```

```
walls.4119                    ! walls file name
p4119.pbd                     ! pbd input file name
```

C.2.2 Walls.4119 File

```
4119 ITTC walls case
    1.000000    7.500000    0.000000    2.000000
    7.600000    0.200000
    5.449084    0.200000
    5.394665    0.200000
    5.340678    0.200000
    5.287190    0.200000
    5.234253    0.200000
    5.181955    0.200000
    5.130387    0.200000
    5.079632    0.200000
    5.029772    0.200000
    4.980885    0.200000
    4.933049    0.200000
    4.886336    0.200000
    4.840815    0.200000
    4.796552    0.200000
    4.753613    0.200000
    4.712055    0.200000
    4.671937    0.200000
    4.633312    0.200000
    4.596230    0.200000
    4.560740    0.200000
    4.526885    0.200000
    4.494707    0.200000
    4.464243    0.200000
    4.435529    0.200000
    4.408596    0.200000
    4.383472    0.200000
    4.360183    0.200000
    4.338752    0.200000
    4.319196    0.200000
    4.301532    0.200000
    4.285772    0.200000
    4.271927    0.200000
    4.260002    0.200000
    4.250000    0.200000
    4.240000    0.200000
    4.230000    0.200000
    4.220000    0.200000
    4.210000    0.200000
    4.200000    0.200000
    4.190000    0.200000
    4.180000    0.200000
    4.170000    0.200000
    4.160000    0.200000
    4.150000    0.200000
    4.140000    0.200000
    4.130000    0.200000
```

4.120000	0.200000
4.110000	0.200000
4.100000	0.200000
4.090000	0.200000
4.080000	0.200000
4.070000	0.200000
4.060000	0.200000
4.050000	0.200000
4.040000	0.200000
4.030000	0.200000
4.020000	0.200000
4.010000	0.200000
4.000000	0.200000
3.250000	0.200000
3.237500	0.200000
3.225000	0.200000
3.212500	0.200000
3.200000	0.200000
3.188016	0.199641
3.176374	0.198600
3.165110	0.196933
3.154251	0.194697
3.143820	0.191947
3.133834	0.188738
3.124304	0.185122
3.115236	0.181149
3.106635	0.176870
3.098497	0.172329
3.090819	0.167569
3.083593	0.162633
3.076809	0.157556
3.070454	0.152374
3.064515	0.147119
3.058977	0.141818
3.053822	0.136499
3.049034	0.131184
3.044594	0.125893
3.040485	0.120644
3.036688	0.115453
3.033185	0.110331
3.029959	0.105290
3.026990	0.100338
3.024263	0.095481
3.021761	0.090723
3.019467	0.086069
3.017367	0.081518
3.015447	0.077072
3.013693	0.072730
3.012093	0.068492
3.010637	0.064357
3.009314	0.060322
3.008113	0.056386
3.007026	0.052547
3.006046	0.048802

3.005163	0.045150
3.004371	0.041587
3.003665	0.038112
3.003037	0.034722
3.002482	0.031414
3.001996	0.028186
3.001573	0.025038
3.001210	0.021965
3.000901	0.018968
3.000645	0.016045
3.000436	0.013194
3.000271	0.010414
3.000148	0.007705
3.000064	0.005067
3.000016	0.002498
3.000000	0.000000
3.000016	-0.002498
3.000064	-0.005067
3.000148	-0.007705
3.000271	-0.010414
3.000436	-0.013194
3.000645	-0.016045
3.000901	-0.018968
3.001210	-0.021965
3.001573	-0.025038
3.001996	-0.028186
3.002482	-0.031414
3.003037	-0.034722
3.003665	-0.038112
3.004371	-0.041587
3.005163	-0.045150
3.006046	-0.048802
3.007026	-0.052547
3.008113	-0.056386
3.009314	-0.060322
3.010637	-0.064357
3.012093	-0.068492
3.013693	-0.072730
3.015447	-0.077072
3.017367	-0.081518
3.019467	-0.086069
3.021761	-0.090723
3.024263	-0.095481
3.026990	-0.100338
3.029959	-0.105290
3.033185	-0.110331
3.036688	-0.115453
3.040485	-0.120644
3.044594	-0.125893
3.049034	-0.131184
3.053822	-0.136499
3.058977	-0.141818
3.064515	-0.147119
3.070454	-0.152374

```
3.076809   -0.157556
3.083593   -0.162633
3.090819   -0.167569
3.098497   -0.172329
3.106635   -0.176870
3.115236   -0.181149
3.124304   -0.185122
3.133834   -0.188738
3.143820   -0.191947
3.154251   -0.194697
3.165110   -0.196933
3.176374   -0.198600
3.188016   -0.199641
3.200000   -0.200000
3.212500   -0.200000
3.225000   -0.200000
3.237500   -0.200000
3.250000   -0.200000
```

Appendix D

Extreme Computations

This chapter represents output from PBD and MTFLOW that may indicate problems with the solution. The solutions are also included to guide the curious reader.

D.1 MTFLOW Spline Problems

Because the input data is splined within MTFLOW, any non-smooth data results in spline interpolation problems. If these problems are ignored, unrealistic flow fields like the one shown in Figure D-1 result.

D.2 PBD Wake Routine Problems

A viscous boundary layer along the downstream hub or casing (duct) can lead to problems. Because the meridional fluid velocity is slower at the hub and tip radii and faster in the propeller jet, the wake will appear to bow in the middle.

The solution is to improve the velocity smearing scheme used within the program *VELCON* such that the meridional velocity field is more nearly uniform.

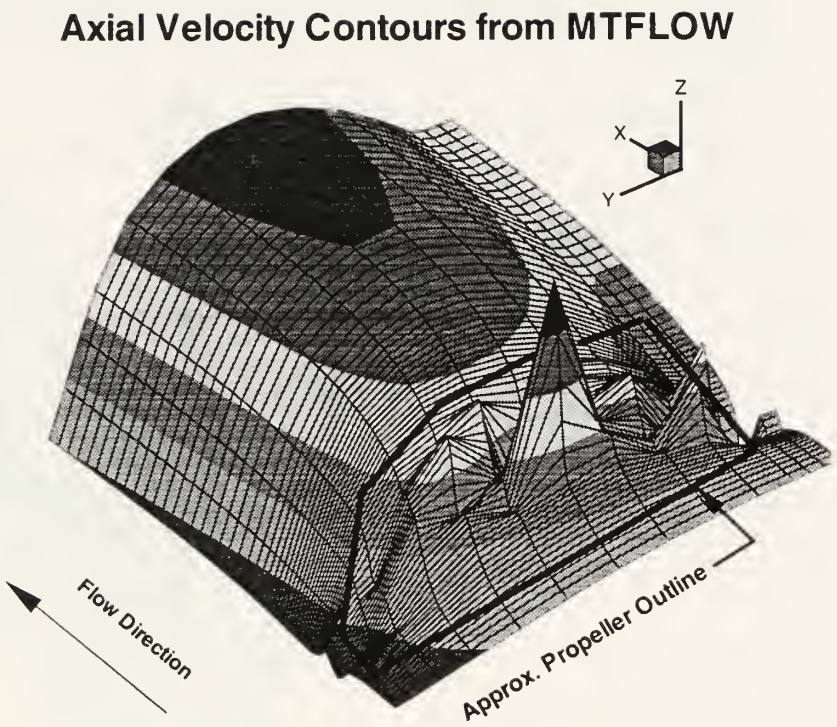


Figure D-1: The flowfield resulting from an MTFLOW input with spline interpolation problems. This physically unrealistic flowfield is handed back to PBD, which solves the blade analysis problem. The results are highly suspect, even though convergence may be achieved.

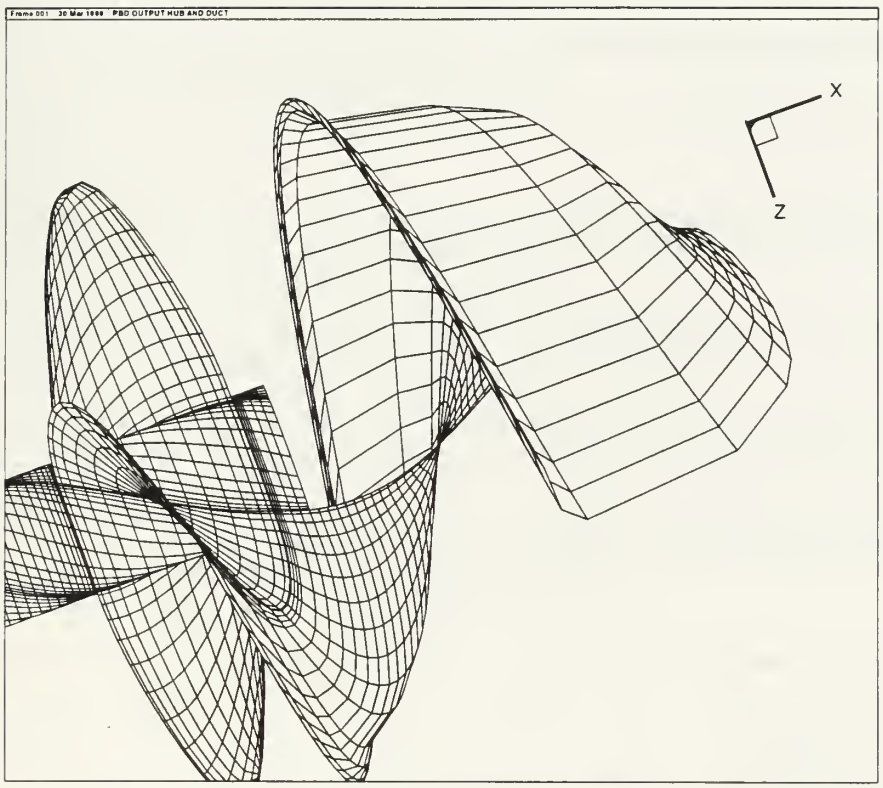


Figure D-2: Wake shot of 3-bladed 4119 with viscous boundary layer on the shaft.

Bibliography

[1] J.L. Allsion, Changben Jiang, J.G. Stricker, M.M. Athavale, J. Kerwin, and T. Taylor. Modern tools for waterjet pump design and recent advances in the field. In *Int'l Conference on Waterjet Propulsion Latest Developments*. The Royal Institution of Naval Architects, October 1998.

[2] William Coney. *A Propeller Lifting Line Model*. PhD thesis, Massachussets Institute of Technology, Dept. of Ocean Engineering, 1989.

[3] M. Drela. *A User's Guide to MTFLOW 1.2*. MIT Fluid Dynamics Research Laboratory, November 1997.

[4] M. Drela and M. Giles. Conservative streamtube soltuion of steady-state euler equations. Technical Report CFDL-TR-83-6, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, November 1983.

[5] David S. Greeley and Justin E. Kerwin. Numerical methods for propeller design and analysis in steady flow. *SNAME Transactions*, 90:415–453, 1982.

[6] T.T. Huang and N.C. Groves. Effective wake : Theory and experiment. Technical Report DTNSRDC 81/033, David W. Taylor Naval Ship Research and Development Center, 1981.

[7] T.T. Huang, N.C. Groves, and G. Belt. Boundary layer flow on an axisymmetric body with an inflected stern. Technical Report DTNSRDC-80/064, David W. Taylor Naval Ship Research and Development Center, 1980. Huang Body 5 Experimental Results and Modified Cebeci Smith Closure Relations for Thick Boundary Layers.

[8] T.T. Huang, N. Santelli, and G. Belt. Stern boundary-layer flow on axisymmetric bodes. Twelfth Symposium on Naval Hydrodynamics, pages 127–157. National Academy of Sciences, 1979.

[9] T.T. Huang, H.T. Wang, N. Santelli, and N.C. Groves. Propeller/stern/boundary-layer interation on axisymmetric bodes: Theory and experiment. Technical Report DTNSRDC 76-0113, David W. Taylor Naval Ship Research and Development Center, December 1976. Huang Bodes 1,2, Streamline curvature with Integral Boundary Layer relations and Granville Wake Relations.

[10] ITTC. 22nd ittc propuelsion committee propeller rans/panel method workshop. International Towing Tank Committee, 1998.

[11] J. L. Kerrebrock. *Aircraft Engines and Bas Turbines*. The MIT Press, second edition, 1996. pages 207-209.

[12] J. E. Kerwin. A vortex lattice method for propeller blade design: Mit-pbd-10 user's manual. Technical Report Report 84-2, MIT Department of Ocean Engineering, January 1984.

[13] J. E. Kerwin. 13.04: Hydrofoils and propellers class notes. Technical report, M.I.T., 1997.

[14] J.E. Kerwin, D.P. Keenan, S.D. Black, and J.G. Diggs. A coupled viscous/potential flow design method for wake-adapted, multi-stage, ducted propulsors using generalized geometry. *SNAME Transactions*, 102:23–56, 1994.

[15] Justin E. Kerwin and Chang-Sup Lee. Prediciton of steady and unsteady marine propeller performance by numerical lifting-surface theory. *SNAME Transactions*, 86:218–253, 1978.

[16] G. P. McHugh, T. E. Taylor, W. M. Milewski, and J.E. Kerwin. Pbd-14.4: A coupled lifting -surface design/analysis code for marine propulsors. Technical report, Depart of Ocean Engineering, Massachusetts Institute of Technology, August 1998.

[17] Gerard P. McHugh. Advances in ducted propulsor analysis using vortex-lattice lifting-surface techniques. Naval engineer's thesis, Massachusetts Institute of Technology, Dept. of Ocean Engineering, 1997.

[18] J.N. Newman. *Marine Hydrodynamics*. MIT Press. 1989.

[19] R.A. Novak. Streamline curvature computing procedures for fluid flow problems. Number paper 66-WA/GT-3, 1966. presented at ASME Winter Annual Meeting.

[20] Michael J. Pierzga. A solution to the direct problem of an open propeller using the streamline curvature method. Technical report, AIAA 78-310, Applied Research Laboratory, Penssylvania State University, 1978.

[21] Airflow Research. Trial fan design.

[22] E.G. Hauptmann R.H. Sabersky, A.J. Acosta. *Fluid Flow: A First Course in Fluid Mechanics*. Macmillan, third edition, 1989.

[23] T. W. Swafford. Analytical approximation of two-dimensional separated turbulent boundary-layer velocity profiles. *AIAA Journal*, 21:923–925, June 1983.

[24] T.E. Taylor. Wake velocity survey using two RANS codes on huang body 1. unpublished analysis, 1998.

[25] F.M. White. *Viscous Fluid Flow*. McGraw-Hill, 2nd edition edition, 1991.